2021/2022

Trabajo de Fin de Grado

Diseño y despliegue de una infraestructura de clúster que ofrece un servicio web basado en *Apache* en alta disponibilidad y con balanceo de carga en *Oracle Linux*

Autor

Danilo Rivero Pérez

Tutores

Dr. Francisco Alexis Quesada Arencibia - Ciencia De La Comp. E Intel. Artificial

Dr. Carmelo Rubén García Rodríguez - Ciencia De la Comp. E Intel. Artificial

Grado en Ingeniería Informática: Las Palmas de Gran Canaria a Julio de 2022.

Agradecimientos

Me gustaría agradecer a todas las personas que han contribuido en mayor o menor medida en el desarrollo de este proyecto final. En especial, a mis dos tutores, Alexis Quesada Arencibia y Carmelo Rubén García Rodríguez, por su disposición en todo momento y prestarme la ayuda necesaria para llevar a cabo este trabajo.

Por otro lado, agradecer a todos mis familiares, en especial a mis padres, ya que han sido una pieza fundamental en mi motivación para lograr realizar este trabajo de fin de grado. Y como no, a mis amigos, que siempre han estado ahí apoyándome en todo momento.

Por último, agradecer al centro tecnológico IUCTC y a todo su personal, por prestarme la ayuda necesaria en cada momento, haciéndome sentir como en casa.

Resumen

Tras el anuncio de la discontinuidad de la distribución de Linux CentOS a partir de su versión 8, en este trabajo se exploran las posibilidades de la distribución Oracle Linux 8.6 para el diseño y despliegue de un clúster que ofrece un servicio web basado en Apache y haciendo uso de un servicio de base de datos, para proporcionar alta disponibilidad y balanceo de carga a la hora de atender las peticiones que puedan realizar sus potenciales clientes. El servicio web a ofrecer en alta disponibilidad y con balanceo de carga se trata del conocido CMS *Wordpress*. Además, el clúster dispone de un servidor externo iSCSI que proporciona un almacenamiento compartido para sus diferentes nodos. Todo el software utilizado es de distribución libre y además, los nodos del clúster estarán virtualizados haciendo uso de la plataforma KVM.

Palabras clave: Clúster, alta disponibilidad, balanceo de carga y virtualización.

Abstract

After the announcement of the discontinuity of the CentOS Linux distribution as of version 8, this paper explores the possibilities of the Oracle Linux 8.6 distribution for the design and deployment of a cluster that offers a web service based on Apache and making use of a database service, to provide high availability and load balance when it comes to meeting the requests that your potential clients may make. The web service to offer in high availability and with load balance is the well-known Wordpress CMS. In addition, the cluster has an external iSCSI server that provides shared storage for its different nodes. All the software used is freely distributed and, in addition, the cluster nodes will be virtualized using the KVM platform.

Keywords: Cluster, high availability, load balancing and virtualization.

Índice de contenido

Índice de ilustraciones

Índice de tablas

1. Introducción

Hoy en día, un requisito importante en los sistemas de información es la alta disponibilidad, ya que ayuda a garantizar el correcto funcionamiento de estos en un período continuado de tiempo. Muchas veces, la tecnología y las redes tienen fallas, por ejemplo, cuando hay un corte en el suministro eléctrico o un error en un servidor. En ciertos sectores, es imprescindible que la red siga funcionando en todo momento o que ciertas máquinas sigan ofreciendo servicios. En otros sectores, un sistema de alta disponibilidad puede ser útil para conservar la información de una manera segura y protegida. Estos sistemas pueden ayudar a minimizar las posibilidades de que se produzcan errores, y si ocurre un suceso imprevisto, como la caída de un servicio por un fallo, pueden ayudar a minimizar el efecto que podría llegar a tener [1].

En el contexto de una organización que posee un servidor en dónde los trabajadores tienen que acceder para hacer uso de distintos programas *software* de su organización o poder acceder a ficheros que tiene almacenado ese servidor. Si ese servidor por el motivo que sea dejara de funcionar, esos trabajadores no podrían desempeñar sus funciones correctamente. Esto supondría un problema grave que podría paralizar el trabajo en la organización. Ahora bien, si se contara con alta disponibilidad en servidores, es decir, hubiera duplicación de hardware, esa duplicación va a permitir que automáticamente haya otro servidor disponible para ofrecer el servicio [2].

Por otro lado, cuando un servidor se vuelve lento debido a la congestión de información, la solución más obvia podría ser la ampliación de la memoria, ampliar el disco duro del servidor o la actualización del procesador del mismo. Pero el tráfico de Internet está en ininterrumpido aumento y lo que se ha expuesto anteriormente sería solamente una solución temporal. Un servidor sobrecargado puede ser perjudicial, pues puede entorpecer e incluso detener algunas áreas de cualquier negocio. Por lo tanto, la mejor opción sería, a largo

plazo, configurar más servidores y distribuir las peticiones de los potenciales clientes entre los servidores configurados, evitando de esta manera una sobrecarga. Esto mejora los tiempos de respuesta, mejora la fiabilidad del sistema y la tolerancia a fallos. Esto último se conoce como "*Load Balancing*" o balanceo de carga [3].

Además, hoy en día son muy utilizadas las infraestructuras de clúster, para mejorar el rendimiento y la disponibilidad. Un clúster, es una arquitectura escalable para el procesamiento distribuido y que está compuesta por varios servidores independientes que funcionan como si fuesen un único servidor; ya que entre todos comparten los recursos de hardware y software ofrecidos, mejorando de esta manera la disponibilidad y la capacidad de rendimiento. Para que todo esto se lleve a cabo, todos los servidores deben estar unidos entre sí mediante una red de alta velocidad que minimiza al mínimo la latencia en el proceso de intercambio de información entre los diferentes servidores del clúster. Esta configuración del sistema clúster permite que, si alguno de los servidores independientes fallara, el resto asumirán su carga para seguir ofreciendo el mismo servicio de manera correcta, proporcionando de esta manera una alta disponibilidad y rendimiento. Existen diferentes tipos de infraestructuras de clúster en función del objetivo perseguido, que se detallarán más adelante en la memoria, pero cabe destacar dos de ellas, clúster de alta disponibilidad y clúster de balanceo de Carga [4].

También, cada vez es más común ver que diferentes organizaciones o empresas, optan por virtualizar sus servidores o máquinas, aprovechando así de una mejor manera los recursos informáticos, reduciendo los costes, ganando más agilidad y eficiencia, permitiendo tener sistemas operativos dentro de otros sistemas operativos y facilitando operaciones tales como: clonar, migrar o copiar.

Por todo ello y una vez anunciada la discontinuidad de la distribución de Linux *CentOS* a partir de su versión 8, la motivación de este Trabajo de Fin de

Grado es explorar las posibilidades de la distribución de Linux denominada "*Oracle Linux"* en su última versión (8.6) para el diseño y despliegue de un clúster que ofrezca un servicio web basado en *Apache* y haciendo uso de un servicio de base de datos, para proporcionar alta disponibilidad y balanceo de carga a la hora de atender las peticiones que puedan realizar sus potenciales clientes. Todo el software a utilizar será de distribución libre, incluido el servicio web basado en Apache y el servicio de base de datos a utilizar. Además, los diferentes nodos del clúster que serán los encargados de ofrecer este servicio web en alta disponibilidad y con balanceo de carga, estarán virtualizados, haciendo uso de la plataforma de virtualización *Kernel-based Virtual Machine* o KVM.

2. Estructura de la memoria

Este documento está organizado en varios apartados/secciones, que se detallan a continuación:

- ***Capítulo I***: Estado actual y Objetivos.

  En este capítulo se mencionará cuál es el estado actual de este trabajo de Fin de Grado, así como la motivación para su realización y los objetivos que se pretenden conseguir con el mismo.

- ***Capítulo II***: Justificación de las competencias específicas cubiertas.

  En este segundo capítulo, se listarán y se justificarán las competencias específicas que han sido cubiertas con la realización de este Trabajo de Fin de Grado.

- ***Capítulo III***: Aportaciones.

  En este tercer capítulo, se justificará la aportación al entorno socio-económico y al entorno técnico/científico que ha supuesto este Trabajo de Fin de Grado.

- ***Capítulo IV***: Normativa y Legislación

  En este capítulo se aclarará la normativa utilizada en el proyecto, describiendo las distintas licencias software utilizadas.

- ***Capítulo V***: Desarrollo del trabajo

  En este capítulo se comentará la metodología aplicada en el transcurso del proyecto y la planificación del mismo.

- ***Capítulo VI***: Virtualización

  En este capítulo se aclarará en que consiste la tecnología de la virtualización, sus tipos y las ventajas de su uso.

- ***Capítulo VII***: Computación en clúster

  En este capítulo se definirá el significado de un clúster, sus componentes y los tipos de clúster existentes.

- ***Capítulo VIII***: Tecnologías principales utilizadas

  En este capítulo se describirá las tecnologías principales usadas en el desarrollo de este trabajo.

- ***Capítulo IX***: Configuración del host anfitrión

  En este capítulo se configurará el sistema anfitrión con una distribución Oracle Linux con el propósito de llevar a cabo el proyecto.

- ***Capítulo X***: Instalación de la plataforma de virtualización KVM

  En este capítulo se procederá a instalar y configurar la plataforma de virtualización KVM con el fin de dejarla operativa para su uso.

- ***Capítulo XI***: Operaciones con máquinas virtuales

  En este capítulo se prueba el funcionamiento de la plataforma KVM realizando operaciones con sus máquinas virtuales.

- ***Capítulo XII***: Recursos de almacenamiento virtual

  En este capítulo se crearán contenedores locales y contendores en red, con el fin de proporcionar un almacenamiento virtual a los sistemas invitados.

- ***Capítulo XIII***: Creación de un bridge en la interfaz física

  En este capítulo se creará un bridge en la interfaz física del sistema anfitrión con el fin de conectar las máquinas virtuales a la misma red que el host anfitrión.

- ***Capítulo XIV***: Diseño y despliegue de la infraestructura de clúster en alta disponibilidad y con balanceo de carga.

  Este será el capítulo principal y más extenso de la memoria, ya que en él se diseña y se despliega una infraestructura de clúster que ofrece alta disponibilidad y balanceo de carga a sus servicios.

- ***Capítulo XV***: Conclusiones y trabajos futuros

  En este capítulo se proporcionará la conclusión final del desarrollo del proyecto y algunos trabajos futuros que se podrían llevar a cabo para su mejora.

- ***Anexos***

  En este apartado se proporcionará el material complementario al desarrollo de la memoria, así como algunas configuraciones realizadas que enriquecen el contenido de la misma.

- ***Fuentes de información***

  En esta sección se encontrarán todas las fuentes bibliográficas consultadas para el desarrollo de esta memoria.

3. Capítulo I: Estado actual y Objetivos

3.1. Estado actual

Tal y como se explica en [5], *CentOS* ha sido durante varios años el referente de los clones de binarios de *Red Hat Enterprise Linux* (RHEL), ya que ofrece exactamente lo mismo que su distribución madre, *Red Hat*, pero sin coste alguno y con un poderoso soporte comunitario. Sin embargo, esto cambió a finales del año 2021, ya que CentOS, en su versión 8, pasó a ser una distribución *rolling release*. Por lo tanto, *CentOs* se focalizará en una nueva distribución denominada *CentOS Stream.* Esto quiere decir que *CentOS* dejará de basarse en las compilaciones estables de RHEL, introduciendo ciertos cambios radicales que afectarán a muchas implementaciones de producción.

Como se ha mencionado anteriormente, este cambio afectará a la versión 8 de la distribución de *CentOS*, que ha sido descontinuado el pasado 31 de diciembre de 2021 para dar paso a *Stream* y servir de esta manera, según el anuncio oficial, *"como la rama upstream (desarrollo) de Red Hat Enterprise Linux."*. Mientras que la versión 7 de *CentOS* seguirá con la programación prevista para su ciclo de vida, por lo que su soporte se mantendrá bajo el mismo modelo hasta el año 2024, año en el que se le dejará de dar soporte a *RHEL 7*.

En la asignatura denominada "*Infraestructuras Tecnológicas para los Sistemas de Información"*, que se imparte en el plan antiguo del Grado en Ingeniería Informática de la ULPGC (Plan 2010), se hace uso de una distribución de Linux *CentOS 7* para la realización de sus prácticas. Esta asignatura, pertenece al plan antiguo (Plan 2010), como se ha comentado anteriormente, pero en el plan nuevo del Grado en Ingeniería informática (Plan 2019) también se imparte, pero con un nombre diferente, el cual es, "*Virtualización y Procesamiento Distribuido*". Estas asignaturas, constan de una serie de prácticas

en dónde se ponen de manifiesto los conocimientos teóricos impartidos en la misma, poniendo en práctica los conocimientos aprendidos acerca de virtualización, operaciones con máquinas virtuales, recursos de almacenamiento virtual, infraestructuras de red y finalizando con el diseño y despliegue de una infraestructura de clúster básico en alta disponibilidad. Todas estas prácticas, como ya se ha mencionado, se llevan a cabo haciendo uso de una distribución de Linux *CentOS* en su versión 7.

Por lo tanto, una vez anunciada y conocida la discontinuidad de la distribución de Linux *CentOS* a partir de su versión 8, explicada anteriormente, se pretende con este Trabajo de Fin de Grado explorar las posibilidades de *Oracle Linux* (en su última versión) para el diseño y despliegue de un servicio web basado en *Apache* que haga uso de un servicio de base de datos, para proporcionar alta disponibilidad y balanceo de carga a la hora de atender las peticiones que puedan realizar sus potenciales clientes. De esta manera, si la exploración de las posibilidades de *Oracle Linux* para la computación en clúster fueran satisfactorias, se podría empezar a utilizar en las asignaturas mencionadas en el párrafo anterior, *Oracle Linux* como distribución para llevar a cabo las prácticas de estas asignaturas en cursos futuros.

3.2. Objetivos

El objetivo final de este trabajo es poner en marcha un servicio web basado en *Apache* que preste sus servicios en alta disponibilidad, es decir, que pueda ejecutarse en varios equipos con el fin de evitar la caída de servicio por un fallo en los sistemas en los que se ejecuta, y que además las peticiones que realicen los clientes potenciales sean atendidas por diferentes equipos y de forma balanceada, garantizando unos tiempos de respuesta adecuados para un servicio de este tipo. Este objetivo será llevado a cabo a través de la distribución de Linux “*Oracle Linux*” y para alcanzarlo, se deberán alcanzar los siguientes objetivos básicos:

- ***O1***: Identificar los elementos *hardware* y *software* requeridos para proporcionar un servicio web basado en *Apache* en alta disponibilidad y con tiempos de respuesta adecuados y uniformes.
- ***O2***: Diseñar un clúster para soportar la alta disponibilidad y el balanceo de carga.
- ***O3***: Instalar los elementos de infraestructura requeridos para desplegar el clúster.
- ***O4***: Instalar los componentes *software* del sistema que proporcionen un clúster en alta disponibilidad y con balanceo de carga.
- ***O5***: Instalación del servicio de base de datos y del servicio web basado en *Apache* en alta disponibilidad y con balanceo de carga.

4. Capítulo II: Justificación de las competencias específicas cubiertas

En [6] se definen las diferentes competencias específicas relativas al Grado en Ingeniería Informática, de las cuáles se justificarán las cubiertas en el desarrollo de este proyecto final.

4.1. TI01

"Capacidad para comprender el entorno de una organización y sus necesidades en el ámbito de las tecnologías de la información y las comunicaciones."

Esta competencia ha sido cubierta, de manera que se ha explorado una nueva posibilidad, como es *Oracle Linux*, para el diseño y despliegue de un clúster que ofrece un servicio web en alta disponibilidad y con balanceo de carga. Ayudando de esta forma a las asignaturas "*Infraestructuras Tecnológicas para los Sistemas de Información" y* "*Virtualización y Procesamiento Distribuido*" impartidas en la ULPGC, a hacer frente a la discontinuidad de *CentOS*.

4.2. TI02

"Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, construir, gestionar, explotar y mantener las tecnologías de hardware, software y redes, dentro de los parámetros de coste y calidad adecuados."

En este trabajo de Fin de Grado se han analizado los diferentes elementos hardware y software a utilizar, y se han integrado en la configuración de la infraestructura de clúster virtualizada que se ha diseñado y se ha desplegado, de la manera más óptima posible. Además, todo el software utilizado ha sido de distribución libre.

4.3. TI04

"Capacidad para seleccionar, diseñar, desplegar, integrar y gestionar redes e infraestructuras de comunicaciones en una organización."

Puesto que en este trabajo se ha diseñado y se ha desplegado una infraestructura de clúster virtualizada, compuesta por un conjunto de nodos interconectados entre sí mediante redes virtuales, esta competencia está más que cubierta.

4.4. TI05

"Capacidad para seleccionar, desplegar, integrar y gestionar sistemas de información que satisfagan las necesidades de la organización, con los criterios de coste y calidad identificados."

En este trabajo se ha diseñado una infraestructura de clúster que ofrece un servicio web en alta disponibilidad y con balanceo de carga, mejorando en gran medida el rendimiento y la calidad del servicio web ofrecido.

4.5. TI06

"Capacidad de concebir sistemas, aplicaciones y servicios basados en tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil."

Se ha puesto en marcha un servicio web basado en *Apache* que presta sus servicios en alta disponibilidad, es decir, que pueda ejecutarse en varios equipos con el fin de evitar la caída de servicio por un fallo en los sistemas en los que se ejecuta, y que además las peticiones que realicen los clientes potenciales sean atendidas por diferentes equipos y de forma balanceada, de forma que se garanticen unos tiempos de respuesta adecuados para un servicio de este tipo.

4.6. TI07

"Capacidad para comprender, aplicar y gestionar la garantía y seguridad de los sistemas informáticos."

Se han asegurado los diferentes sistemas informáticos virtualizados de la infraestructura de clúster de la mejor forma posible. Se ha hecho uso del servicio de seguridad "*SELinux*" para permitir que los administradores tengan un mayor control acerca de los usuarios que pueden acceder a los sistemas informáticos, controlando el acceso a las aplicaciones, procesos, etc. Además, el servicio web ofrecido en alta disponibilidad y con balanceo de carga, es proporcionado por una infraestructura compuesta de varios nodos que están interconectados entre sí haciendo uso de redes virtuales que operan de la manera más segura posible. También, se hace uso de un cortafuegos para poner en marcha el servicio web y configurar determinadas funcionalidades de la plataforma de virtualización KVM.

5. Capítulo III: Aportaciones

5.1. Entorno socio-económico

Con la realización de este trabajo, se ha conseguido explorar una nueva plataforma para el diseño y despliegue de un servicio web basado en Apache que haga uso de un servicio de base de datos, para proporcionar alta disponibilidad y balanceo de carga a la hora de atender las peticiones que puedan realizar sus potenciales clientes. Esta plataforma se trata de la distribución de GNU/Linux llamada *Oracle Linux* y que tras su exploración puede ser de gran ayuda para que el profesorado de las asignaturas "*Infraestructuras Tecnológicas para los Sistemas de Información" y* "*Virtualización y Procesamiento Distribuido*" impartidas en la ULPGC, opten por usarla en las prácticas de estas, tras la conocida discontinuidad de *CentOS*, distribución utilizada hasta ahora. Esto aporta mucho al entorno social de la Universidad de Las Palmas de Gran Canaria (ULPGC).

Además, la distribución *Oracle Linux* se puede descargar y usar de una manera totalmente gratuita, asimismo como todo el software utilizado en este trabajo, ya que la ULPGC tiene un convenio con Oracle que permite el uso de sus recursos con fines académicos. Todo esto, supone una gran ventaja con respeto al entorno económico. También, como la infraestructura del clúster es virtual, se consiguen varias ventajas como: aumento de la productividad, la eficiencia, la agilidad, la capacidad de respuesta, mejor aprovechamiento de los recursos y reducción de costes.

5.2. Entorno técnico/científico

Como ya se ha mencionado anteriormente, con este trabajo se ha conseguido ofrecer un servicio web en alta disponibilidad y con balanceo de carga, mejorando el rendimiento en todos sus sentidos del servicio ofrecido.

Además, tras la realización de este proyecto se ha conseguido explorar una nueva posibilidad, similar y compatible con Red Hat, como es la distribución *Oracle Linux*. Con esta distribución, se podrá hacer frente a la discontinuidad conocida de *CentOS* a partir de su versión 8, y poder seguir impartiendo de la manera óptima algunas asignaturas de la ULPGC, aportando en gran medida al entorno técnico/científico.

6. Capítulo IV: Normativa y Legislación

6.1. Licencias Software

Tal y como se menciona en [7], las licencias software son contratos en los que los usuarios que hacen uso de ellas aceptan unos términos y condiciones del fabricante para poder hacer uso del software. Además, si el software es de distribución libre o es propietario, los términos y condiciones de las licencias serán más o menos restrictivos.

Por lo cual, las licencias son los permisos que un fabricante suministra para la distribución, uso y modificación del software. Además, estas licencias pueden estar limitadas a periodos de tiempo, territorio en donde se aplica, etc.

6.1.1. Licencia GNU (GPL)

La *General Public License* (GNU GPL) *[8]* es una licencia de derecho de autor creada originalmente por Richard Stallman, fundador de la *Free Software Foundation* (FSF) para el proyecto GNU. Esta licencia es muy utilizada en el mundo del software libre y en el *Open Source y* proporciona a los usuarios la libertad para usar, estudiar, compartir y personalizar el software. Además, el propósito de esta licencia es doble: por un lado, busca declarar que el software cubierto por la licencia es libre y por otro lado, busca protegerlo (mediante

copyleft) de intentos de apropiación que restrinjan esas libertades a nuevos usuarios cada vez que se distribuya, se modifique o se amplie. Bajo esta licencia se han usado el Sistema Operativo *Oracle Linux*, la plataforma de virtualización *KVM*, el cortafuegos *FirewallD*, el gestor de paquetes *DNF*, el sistema de archivos compartido *OCFS2*, el software para el equilibrio de carga *HAProxy*, el módulo de seguridad *SELinux*, el sistema de gestión de contenidos *Wordpress* y el sistema gestor de bases de datos *MariaDB*, usados durante el transcurso de este proyecto.

6.1.2. Licencia de Apache

La *Apache License [9]*, es una licencia de sofware libre permisiva que fue creada por la *Apache Software Foundation* (ASF). Esta licencia requiere la conservación del aviso de derecho de autor y el descargo de responsabilidad y además, no necesita la redistribución del código fuente cuando se distribuyen versiones modificadas, por lo que no es una licencia *copyleft*. Bajo esta licencia se ha usado el servidor web Apache utilizado en el transcurso de este proyecto.

6.1.3. Licencia de PHP

La licencia PHP *[10],* es la licencia bajo la que se divulga el lenguaje de programación PHP. Según la *Free Software Foundation* (FSF), esta licencia es de software libre no *copyleft* y una licencia de código abierto (*Open Source*) según la *Open Source Initiative*. Además, esta licencia no es compatible con la licencia GPL debido a la restricción en el uso del término “PHP”. Bajo esta licencia se ha usado el lenguaje de programación PHP utilizado en el transcurso de este proyecto.

6.1.4. Licencia de Oracle

Como ya se ha mencionado anteriormente, la ULPGC dispone de un convenio con Oracle con el fin de poder utilizar sus recursos (distribución y documentación) con fines académicos. Bajo esta licencia se ha usado la distribución de Oracle Linux.

6.1.5. Tabla-Resumen de las licencias software

Tecnologías usadas	Licencia GNU GPL	Licencia de Apache	Licencia de PHP	Licencia de Oracle
Oracle Linux	X			X
Apache		X		
MariaDB	X			
PHP			X	
Wordpress	X			
SELinux	X			
HAProxy	X			
FirewallD	X			
KVM	X			
DNF	X			
OCFS2	X			

Tabla 1: Licencias software utilizadas

7. Capítulo V: Desarrollo del trabajo

7.1. Metodología aplicada

El proyecto se basa en la instalación y despliegue de un servicio basado en tecnología Web, concretamente un servicio web basado en *Apache* que haga uso de un servicio de base de datos (MariaDB), ejecutándose en una infraestructura que proporciona alta disponibilidad y balanceo de carga en los nodos de cómputo. Para ello, se ha utilizado el soporte de computación en clúster adecuado y soportado por Oracle Linux.

Por tanto, se trata de un proyecto propio de un arquitecto de sistemas y de un administrador de sistemas con un grado de complejidad reseñable. Para ello el proyecto se desarrollará por etapas, que serán:

- **E1**: Análisis de requerimientos del servicio web basado en Apache y del servicio de base de datos a utilizar.
- **E2**: Análisis de requerimientos de la plataforma Oracle Linux para la computación en clúster.
- **E3**: Diseño del clúster.
- **E4**: Instalación de los elementos de infraestructura para soportar la configuración clúster diseñada (nodos de cómputo, nodos de almacenamiento e infraestructura de red)
- **E5**: Instalación del clúster.
- **E6**: Instalación del servicio web basado en Apache en el clúster y de la base de datos.
- **E7**: Afinamiento de los parámetros de configuración para garantizar un rendimiento apropiado.

7.2. Planificación inicial

A continuación, se muestra una tabla con la planificación inicial llevada a cabo para el desarrollo de este proyecto de fin de grado:

Fases	Duración estimada (horas)	Tareas
Estudio previo / Análisis	30	Tarea 1.1: Estudio de los sistemas software para desplegar un servicio web basado en *Apache y de un servicio de base de datos*.
		Tarea 1.2: Estudio de los componentes de *Oracle Linux* para proporcionar alta disponibilidad con balanceo de carga.
Diseño / Desarrollo / Implementación	190	Tarea 2.1: Diseño del clúster.
		Tarea 2.2: Instalación de los elementos de infraestructura del clúster.
		Tarea 2.3. Instalación del clúster.
		Tarea 2.4. Instalación del servicio web basado en *Apache y del servicio de base de datos*.
Evaluación / Validación / Prueba	20	Tarea 3.1: Pruebas para verificar la alta disponibilidad del servicio.
		Tarea 3.2: Pruebas para verificar que las peticiones se atienden de manera balanceada.
Documentación / Presentación	60	Tarea 4.1: Desarrollo de la documentación de la memoria del trabajo de fin de título y manuales de usuario.
		Tarea 4.2: Preparación de presentación y defensa del TFT.

Tabla 2: Planificación inicial del TFG

Además, el tiempo invertido en realizar las distintas fases de este trabajo de fin de grado se ha ajustado en gran medida a la planificación inicial estimada, por lo que no se ha tenido que hacer ningún ajuste en esta planificación inicial.

8. Capítulo VI: Virtualización

8.1. ¿Qué es la virtualización?

La virtualización es una tecnología que permite que varias máquinas virtuales se ejecuten sobre una maquina física con el objetivo de aprovechar lo máximo posible los recursos del sistema, aumentando de esta manera su rendimiento. Es importante destacar que a cada una de las máquinas virtuales se le pueden asignar unos recursos específicos (memoria, unidades de almacenamiento, procesador, redes …) y ejecutan una copia propia de sistema operativo (Windows, Linux …) [11].

8.2. Ventajas de la virtualización

Algunas de las ventajas de la virtualización [12] son:

- Con la virtualización se puede llegar a cargas superiores al 50%, ya que se pueden simular varios servidores que ofrezcan varios servicios en una misma máquina.
- Permite una rápida y fácil incorporación de nuevos recursos para los servidores virtualizados (máquinas virtuales).
- Permite una administración centralizada y simple de los servidores virtualizados.
- Facilidad para la creación de entornos de test, ya que permite probar software en distintas configuraciones hardware sin tener que disponer de él de manera física.
- Proporciona aislamiento, debido a que los problemas o fallas de los que puede sufrir una máquina virtual no afecta a las otras máquinas.
- Reduce el hardware necesario y reduce los costes de software en determinados casos.

- Permite la migración en caliente de máquinas virtuales de un servidor físico a otro.

8.3. Tipos de virtualización

Tal y como se precisa en [13], se puede clasificar la virtualización según lo que se quiere virtualizar en cinco tipos.

8.3.1. Virtualización de la plataforma

También conocida como virtualización de hardware, este tipo de virtualización consiste en que se tienen varias máquinas virtuales que son softwares huésped ejecutado sobre un software especial denominado Hipervisor o VMM (*Virtual Machine Monitor*). Esta VMM crea una capa de abstracción del hardware de la máquina física y se la ofrece al sistema operativo de la máquina virtual o *guest*.

8.3.2. Virtualización de recursos

Según el tipo de recurso que se quiera virtualizar se dispone de:

- **Virtualización de almacenamiento**: Sistemas RAID, LVM, SAN y NAS.
- **Virtualización de memoria**: Memoria virtual.
- **Virtualización de red**: VPN, VLAN, etc.
- **Virtualización de interfaces de red**: La agregación de enlaces simula un dispositivo de red único con un gran ancho de banda utilizando varias interfaces de red independientes.
- **Virtualización de recursos de computación:** Clúster de computación.

8.3.3. Virtualización de entornos software en ejecución

Dependiendo del entorno de ejecución que se virtualice se tiene:

- **Virtualización a nivel de sistema operativo**: El kernel del sistema operativo permite la ejecución de forma paralela de instancias de espacio de usuario aisladas unas de otras. Todas las instancias comparten el kernel del sistema operativo subyacente y en cada instancia el usuario puede instalar una o más aplicaciones. Ejemplos: Virtualización basada en Linux, OpenVZ, Solaris Containers, etc.

- **Virtualización de escritorio:** Implementan un escritorio como servicio basándose en una estructura cliente-servidor, de manera que se permite tener una administración centralizada y en entornos de trabajo estandarizados cabe la posibilidad de reducir los costes de gestión y mantenimiento. Ejemplos: VNC, RDP, VMware ACE, etc.

8.3.4. Virtualización de aplicaciones

Este tipo de virtualización consiste en utilizar aplicaciones que aparentemente se ejecutan en local. Por ejemplo, con aplicaciones remotas, en donde las aplicaciones se ejecutan en un servidor remoto y el cliente se conecta a ellas a través de protocolos de visualización como *VNC* o *RDP*. Otro ejemplo, sería el *Streaming de aplicaciones*, en donde las aplicaciones se ejecutan en local, pero ciertos componentes se descargan de manera que es posible funcionar sin tener conexión a la red. De esta manera, únicamente las partes esenciales del código de una aplicación deben instalarse en el equipo informático. Ejemplos: VMware ThinApp, Microsoft Application Virtualization, etc.

8.3.5. Virtualización de datos

Consiste en integrar datos de diferentes fuentes, en distintas localizaciones y formatos, sin tener datos repetidos, para construir una capa de datos virtuales que facilite la provisión de servicios de datos unificados para dar soporte a distintas aplicaciones y usuarios. Para ello proporciona una capa de abstracción y una capa de servicios de datos. Ejemplos: Oracle, DB2, SQL Server, Impala, Sharepoint, etc.

9. Capítulo VII: Computación en clúster

9.1. ¿Qué es un clúster?

Es la conexión entre dos o más equipos informáticos con el objetivo de mejorar el rendimiento de los sistemas en la ejecución de distintas tareas. Cada equipo informático recibe el nombre de “nodo”. Con esto, los equipos informáticos actúan dentro de un único sistema, trabajando juntos en el procesamiento, análisis e interpretación de datos e información y realizando tareas de forma simultánea [14].

9.2. Componentes de un clúster

Un sistema clúster está conformado por una serie de componentes [15, p.7] tales como:

- **Nodos**: Equipos informáticos o servidores que conforman el clúster.
- **Sistemas Operativos**: Por ejemplo. Linux, Windows, etc.
- **Infraestructura de red de alto rendimiento**: Para minimizar la latencia en el proceso de intercambio de información entre los diferentes servidores del clúster.
- **Protocolos de comunicación y servicios**: TCP/IP y UDP/IP
- **Middleware**: Componente que proporciona una imagen única del sistema.

- **Entorno de desarrollo de aplicaciones:** En donde se ejecutan las aplicaciones.
- **Aplicaciones:** Programas software del clúster.

9.3. Tipos de clúster

En función del objetivo perseguido existen cuatro tipos de clúster [15, p.9]:

- **Alto rendimiento**: Se utiliza para realizar tareas de alto rendimiento garantizando sus altas prestaciones. Se ejecutan el mayor número de procesos por unidad de tiempo.
- **Alta disponibilidad**: Se utiliza para garantizar la disponibilidad de los servicios prestados por el clúster. Si un nodo cae, por alguna falla, otro nodo cogerá el testigo y los servicios seguirán en funcionamiento de forma natural.
- **De balanceo de carga**: Es muy parecido a un clúster de alta disponibilidad, pero con un añadido más. Este añadido tiene que ver con la distribución de las peticiones que reciben los nodos del clúster, distribuyendo las mismas entre los distintos nodos con el fin de no sobrecargar a ningún nodo, garantizando mejores tiempos de respuesta y mejorando la calidad del servicio a ofrecer.
- **Almacenamiento distribuido**: Se utiliza para almacenar datos en más de un nodo con el fin de ofrecer una mayor flexibilidad y rendimiento a la ahora de manejar y acceder a los mismos. Incrementando de esta manera la seguridad de la información que se maneja.

10. Capítulo VIII: Tecnologías principales utilizadas

10.1. Oracle Linux

Distribución de Linux que proporciona un entorno operativo seguro y de buen rendimiento, ofreciendo herramientas de virtualización, automatización, gestión y computación nativa de la nube. Además, *Oracle Linux* es una alternativa 100% compatible a *Red Hat Enterprise Linux* y *CentOS*. Además, *Oracle Linux* en su versión 8 posee las mismas novedades que *Red Hat Enterprise Linux 8.0* (RHEL8). En este proyecto se ha trabajado con esta distribución en su última versión hasta la fecha de finalización de la escritura de este documento (Julio 2022), *Oracle Linux* 8.6 [16].

10.2. KVM

KVM – *Kernel-based Virtual Machine* es una solución de virtualización completa de código abierto para sistemas operativos Linux en hardware x86 que contiene extensiones de virtualización *Intel VT* o *AMD-V*. KVM, usa un hypervisor de Tipo II o no nativo, ya que se ejecuta sobre el S.O de la máquina host anfitrión. Esta plataforma de virtualización consiste en un módulo de kernel cargable (*kvm.ko*) que proporciona la infraestructura de virtualización central y un módulo específico del procesador (*kvm-intel.ko* o *kvm-amd.ko*). Con KVM, se tiene la posibilidad de ejecutar varias máquinas virtuales que ejecuten un Sistema Operativo, cada una de ellas con su hardware virtualizado privado. Por lo cual, esta será la plataforma de virtualización a utilizar en el transcurso de este trabajo [17].

10.3. DNF

La utilidad que se proporciona para administrar el software en Oracle Linux 8 se basa en el gestor de paquetes DNF – *Dandified Yum.* En Oracle Linux

8 se puede usar la utilidad DNF para instalar o actualizar paquetes y sus dependencias, por lo que en este proyecto se usará esta utilidad para el manejo de los paquetes y el software. Este gestor de paquetes proporciona mejoras significativas en la funcionalidad y en el rendimiento en comparación con la utilidad tradicional YUM. Además, posee una gran cantidad de características nuevas, incluida la compatibilidad con contenido modular y una API más estable. DNF es compatible con YUM v3, cuando se utiliza desde la línea de comandos, ya que los comandos *yum* y *dnf* son intercambiables [18].

10.4. Galera Cluster

Galera Cluster es un clúster multiprimario virtualmente síncrono para MySQL, XtraDB y MariaDB (Base de datos usada en este proyecto). Esta tecnología está disponible solamente en Linux y sólo es compatible con el motor de almacenamiento InnoDB. *Galera Cluster* dispone de una serie de características como: permite la replicación síncrona, permite leer y escribir en cualquier nodo del clúster, posee una topología multiprimaria activa-activa, permite la replicación en paralelo real, control de membresía automático, alta disponibilidad, etc. En este proyecto se usará *Galera Cluster* para construir y gestionar un clúster de bases de datos MariaDB, consiguiendo así una replicación síncrona y alta disponibilidad en la base de datos a utilizar [19].

10.5. HAProxy

HAProxy es un proxy inverso gratuito, rápido y fiable que ofrece alta disponibilidad, balanceo de carga y proxy para aplicaciones basadas en TCP y HTTP. Esta tecnología es muy adecuada para sitios web con un alto tráfico. Además, es muy utilizada en varias distribuciones de Linux y por ello se usará a lo largo de este trabajo de fin de grado para ofrecer alta disponibilidad y balanceo de carga al servicio web *Apache* ofrecido y a la base de datos *MariaDB [20]*.

10.6. Apache

Apache es uno de los servidores web *HTTP* de código abierto más populares del planeta tierra. Es un componente clave en la pila de software *LAMP* (compuesta por Linux, MySQL, PHP y Apache) y para el despliegue del servicio web. En este proyecto, se usará *Apache* para ofrecer en alta disponibilidad y con balanceo de carga el servicio de gestión de contenidos *Wordpress [21]*.

10.7. MariaDB

Es un servidor de bases de datos de código abierto basado en la tecnología MySQL que posee una gran velocidad, robustez y múltiples motores de almacenamiento. *MariaDB* es una base de datos relacional que transforma los datos en información estructurada y proporciona una interfaz SQL para el acceso a los datos. *MariaDB* será el sistema de gestión de bases de datos a usar en este trabajo [22].

10.8. PHP

Es un lenguaje de programación de código abierto muy conocido y usado en entornos web y que además puede ser incrustado en HTML. *PHP* se ejecuta en el servidor, genera *HTML* y se lo envía al cliente. En este trabajo, al utilizar el CMS *Wordpress*, se necesitará tener instalado *PHP* [23].

10.9. Wordpress

Es un sistema de gestión de contenidos web de código abierto (también conocido como CMS o *content management system*), que permite publicar contenido en la web de manera sencilla y rápida. *Wordpress* en el líder absoluto a nivel mundial en cuanto a la creación de sitios webs se refiere, ya que es la tecnología más utilizada. En el desarrollo de este trabajo se ofrecerá en alta

disponibilidad y con balanceo de carga un servicio web realizado con este CMS [24].

10.10. iSCSI

Es un protocolo de red de área de almacenamiento alternativo a la *Fibre Channel* (FC) tradicional que utiliza tecnología de almacenamiento IP y que define cómo se transfieren los datos entre los sistemas host y los dispositivos de almacenamiento. *iSCSI* permite el transporte de datos *SCSI* a nivel de bloque entre el iniciador i*SCSI* y el destino de almacenamiento a través de redes *TCP/IP*. En este proyecto se configurará y se usará un servidor externo *iSCSI* virtualizado que proporcionará un disco compartido a los nodos del clúster con el fin de tener un almacenamiento compartido y distribuido [25].

10.11. SELinux

Security Enhanced Linux (o más conocido como *SELinux*) es un servicio de seguridad del sistema que permite que los administradores tengan un mayor manejo sobre los usuarios que puedan acceder a los mismos. *SELinux* implementa el Control de Acceso Obligatorio (*MAC*), donde cada proceso y recurso del sistema tiene una etiqueta de seguridad especial denominada *SELinux context*. Los contextos de SELinux funcionan como identificadores que abstraen los detalles a nivel de sistema y se centran en las propiedades de seguridad de la entidad. SELinux puede actuar en tres modos: *disabled* (la política de SELinux no se aplica), *enforcing* (se aplica la política de SELinux en su modo más restrictivo) o *permissive* (la política de SELinux no se aplica, pero se obtienen mensajes de aviso). A lo largo del trabajo se usará la mayor parte del tiempo esta arquitectura de seguridad en su modo más restrictivo (*enforcing*) [26].

10.12. FirewallD

Demonio de servicio de cortafuegos que proporciona un cortafuegos dinámico y personalizable basado en el host con una interfaz *D-Bus*. *FirewallD* permite la creación, eliminación y modificación de reglas para controlar el tráfico. Además, este demonio utiliza los conceptos de *zones* (conjuntos predefinidos de reglas) y *services*, para simplificar la gestión del tráfico. En este trabajo se controlará y se filtrará el tráfico haciendo uso de este demonio en su forma activa en cualquier máquina (*host* y *guests*) [27].

10.13. OCFS2

Oracle Cluster File System o más conocido como *OCFS*, es un sistema de archivos distribuido que es utilizado en clústeres de servidores de sistemas GNU/Linux y que ha sido desarrollado por *Oracle Corporation*. *OCFS2*, tiene muchas semejanzas con el sistema de archivos *GFS2* nativo de *Red Hat*. Además, en este proyecto se trabajará con la versión dos de este sistema de archivos de discos compartidos (*OCFS2*), ofreciendo más funcionalidades que en su versión uno [28].

11. Capítulo IX: Configuración del Host Anfitrión

11.1. Instalación del Sistema Operativo

En primer lugar, se comenzará instalando el sistema Operativo en el host anfitrión. El host anfitrión del que se dispone, en este caso, se trata de un ordenador con 16GiB de RAM y con un procesador i7 alojado en el Instituto Universitario de Ciencias y Tecnologías Cibernéticas (*IUCTC*).

Para proceder con la instalación, se ha descargado de la página oficial de *Oracle* [29] la imagen ISO correspondiente a la última versión hasta el día de hoy de la distribución *Oracle Linux*, la cual corresponde a la 8.6.

Acto seguido, se ha usado la herramienta *balenaEtcher* [30] para crear una unidad flash USB con la imagen ISO de *Oracle Linux 8.6*. Una vez quemada la imagen ISO en la unidad USB, se ha insertado el USB en el host anfitrión y se ha procedido a realizar los ajustes y configuraciones necesarios en la BIOS del sistema para que se arranque con la unidad flash USB.

Tras arrancar el host anfitrión con la unidad USB mencionada anteriormente, tomará el control el instalador del sistema operativo correspondiente. En el Anexo I se detallan los pasos de la instalación de Oracle Linux 8.6 en el host anfitrión.

11.2. Ajustes del Sistema Anfitrión

En este apartado, se realizarán los ajustes en el sistema anfitrión que se especifican: Actualizar el sistema, cambiar el nombre de dominio del host anfitrión y deshabilitar la utilidad *NetworkManager*.

11.2.1. Actualización del Sistema Anfitrión

Es muy recomendable tener siempre actualizado el sistema, para disponer de la última versión de los paquetes software y de las últimas novedades, por lo que se procederá a actualizar el sistema anfitrión a través de línea de comandos de la siguiente manera:

```
$ dnf upgrade
```

11.2.2. Nombre de dominio del sistema anfitrión

Acto seguido, se procederá a cambiar el nombre de dominio completamente cualificado (*FQDN*) en el host anfitrión, con el fin de identificar al sistema anfitrión mediante un nombre descriptivo y poder resolver su dirección IP asociada al mismo. Para ello, se usará la orden "*hostnamectl"* como se menciona en [31]:

```
$ hostnamectl set-hostname equipo3.tfg.com
```

Para que los cambios surtan efecto se tendrá que reiniciar el sistema:

```
$ reboot
```

Además, se puede comprobar que el nombre de dominio ha sido correctamente modificado haciendo uso de la orden *"hostname -f*" o bien observando el contenido del fichero "*/etc/hostname"*

11.2.3. Deshabilitación de *NetworkManager*

Además, se deshabilitará la utilidad para simplificar el uso de redes en Linux denominada *NetworkManager,* con el objetivo de configurar manualmente las interfaces de red del sistema anfitrión mediante los scripts de configuración de red correspondientes. Para ello, en primera instancia se debe instalar los scripts para procesar la configuración de la red sin necesidad de usar *NetworkManager*, de la siguiente forma:

```
$ dnf -y install network-scripts
```

Tras ello, se debe activar y configurar para que se inicie en el arranque del sistema el servicio “network”:

```
$ systemctl start network
```

```
$ systemctl enable network
```

Por último, se parará y se deshabilitará el servicio *NetworkManage*r, ya que la configuración de red del host anfitrión será gestionada sin usar *NetworkManager*:

```
$ systemctl stop NetworkManager
```

```
$ systemctl disable NetworkManager
```

Se puede comprobar que el servicio *NetworkManager* no está en funcionamiento ejecutando la orden que aparece en la siguiente ilustración (Ver ***Ilustración 1***):

```
root@equipo3 ~ systemctl status NetworkManager
● NetworkManager.service - Network Manager
   Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: man:NetworkManager(8)

abr 28 09:19:30 equipo3.tfg.com systemd[1]: Stopping Network Manager...
abr 28 09:19:30 equipo3.tfg.com NetworkManager[907]: <info>  [1651133970.4059] caught SIGTERM, shutting down normally.
abr 28 09:19:30 equipo3.tfg.com NetworkManager[907]: <info>  [1651133970.4221] dhcp4 (enp0s25): canceled DHCP transaction
abr 28 09:19:30 equipo3.tfg.com NetworkManager[907]: <info>  [1651133970.4222] dhcp4 (enp0s25): state changed bound -> terminated
abr 28 09:19:30 equipo3.tfg.com NetworkManager[907]: <info>  [1651133970.4224] manager: NetworkManager state is now CONNECTED_SITE
abr 28 09:19:30 equipo3.tfg.com NetworkManager[907]: <info>  [1651133970.4228] device (virbr0): bridge port virbr0-nic was detached
abr 28 09:19:30 equipo3.tfg.com NetworkManager[907]: <info>  [1651133970.4228] device (virbr0-nic): released from master device virbr0
abr 28 09:19:30 equipo3.tfg.com NetworkManager[907]: <info>  [1651133970.4256] exiting (success)
abr 28 09:19:30 equipo3.tfg.com systemd[1]: NetworkManager.service: Succeeded.
abr 28 09:19:30 equipo3.tfg.com systemd[1]: Stopped Network Manager.
root@equipo3 ~
```

Ilustración 1: Estado de la utilidad NetworkManager

12. Capítulo X: Instalación de la plataforma de virtualización KVM

Para la preparación del sistema anfitrión e instalación de la plataforma de virtualización KVM, se ha seguido el manual accesible en [32].

12.1. Requerimientos del sistema

Antes de instalar *KVM* en *Oracle Linux*, se deben tener en cuenta algunas pautas y recomendaciones a tener en cuenta en el sistema anfitrión. Estos requisitos tienen que ver con las funciones de virtualización de la CPU del host anfitrión, las proporciones de CPU virtual a CPU host, la memoria reservada, espacio de almacenamiento en disco, la compatibilidad del host, etc. Además, los sistemas operativos invitados también tienen una serie de requisitos y compatibilidades. Se pueden ver con más detalles estos requerimientos mencionados en el Anexo II.

12.2. Paquetes de virtualización

Una vez estudiados y cumplimentados los requisitos del sistema, se estudiarán los paquetes de virtualización que proporciona Oracle Linux para trabajar con KVM. Los paquetes mínimos necesarios para un host de virtualización son:

- **libvirt**: Proporciona una interfaz para *KVM* y el demonio *libvirtd* que es el encargado de administrar las máquinas virtuales *guest's*.
- **qemu-kvm:** Paquete que instala el emulador *QEMU* para la virtualización de hardware para permitir que las máquinas virtuales puedan tener acceso a la CPU del *host* anfitrión y otros recursos.

- **virt-install:** Proporciona utilidades de línea de comandos para crear máquinas virtuales *guest's*.
- **virt-viewer:** Proporciona una utilidad gráfica para acceder a la consola gráfica de una máquina virtual *guest*.

Además, como alternativa a la instalación de paquetes de virtualización de forma individual, se pueden instalar grupos de paquetes y como se está usando un entorno gráfico de escritorio en el host anfitrión, se instalarán cuatro grupos de paquetes para tener una plataforma de virtualización KVM completa (Ver Tabla X):

Grupo de paquetes	Descripción
virtualization-hypervisor	Proporciona lo básico para administrar las máquinas virtuales
virtualization-client	Clientes para instalar y administrar instancias de virtualización
virtualization-platform	Proporciona una interfaz para acceder y controlar huéspedes y contenedores virtualizados
virtualization-tools	Proporciona herramientas para la gestión de imágenes virtuales desconectadas

Tabla 3: Grupos de paquetes de virtualización para KVM

12.3. Repositorios para el software de virtualización

Antes de proceder con la instalación de todas las utilidades y el software que compone a la plataforma de virtualización de KVM, se debe tener instalado el paquete "ora*clelinux-release-el8*" en su versión más reciente. Para ello, se asegura esto último instalando el paquete:

```
$ dnf -y install oraclelinux-release-el8
```

Y se habilitan los dos repositorios que proporcionan los paquetes de virtualización:

```
$ dnf config-manager --enable ol8_appstream ol8_kvm_appstream
```

Nota: Se pueden ver los repositorios habilitados e inhabilitados ejecutando la orden: "*dnf repolist all".*

12.4. Instalación de los grupos de paquetes de virtualización para KVM

Una vez que se han cumplimentado los requisitos del sistema, se han seleccionado los grupos de paquetes de virtualización a instalar y se han habilitado los repositorios que los contienen, se procederá con la instalación de los grupos de paquetes seleccionados anteriormente para finalizar y dejar en funcionamiento la plataforma de virtualización KVM. Para ello, se hará uso del gestor de paquetes "*dnf*" y una de sus opciones, como es *"--setopt*", para instalar los paquetes obligatorios (*mandatory*), los paquetes por defecto (*default*) y los paquetes opcionales (*optional*):

```
$ dnf -y groupinstall "Virtualization Hypervisor" --
setopt=group_package_types=mandatory,default,optional
```

```
$ dnf -y groupinstall "Virtualization Client" --
setopt=group_package_types=mandatory,default,optional
```

```
$ dnf -y groupinstall "Virtualization Platform" --
setopt=group_package_types=mandatory,default,optional
```

```
$ dnf -y groupinstall "Virtualization Tools" --
setopt=group_package_types=mandatory,default,optional
```

Para ver con más detalle los paquetes que se han instalado en cada grupo de paquetes, se puede consultar el Anexo III.

12.5. Módulos de KVM y demonio *libvirtd*

Tras la instalación de todos los paquetes necesarios para el correcto funcionamiento de la plataforma KVM, es recomendable reiniciar el host anfitrión (*reboot*). Una vez reiniciado el sistema, se debe verificar que los módulos del núcleo que dan soporte a KVM se han cargado correctamente durante el arranque, para ello se usará la orden "*lsmod*" (Ver ***Ilustración 2***):

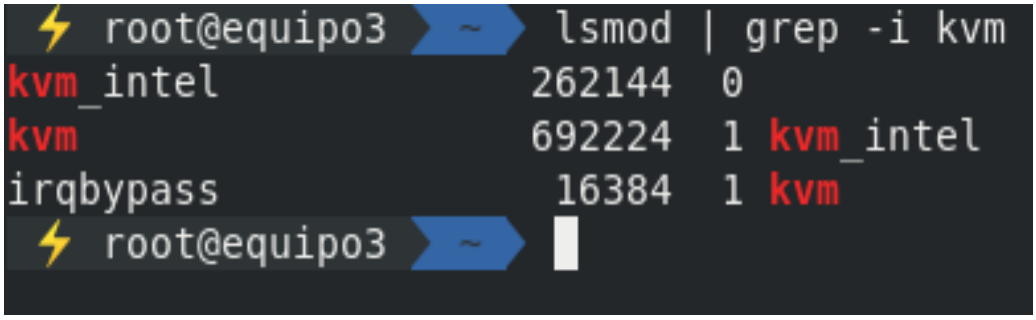

Ilustración 2: Módulos del núcleo que dan soporte a KVM

Además, se debe comprobar que el demonio *libvirtd*, responsable de las invocaciones a las funciones de la librería, manejo de las máquinas *guests* y control del hypervisor, esté activo, como se muestra en la ***Ilustración 3***.

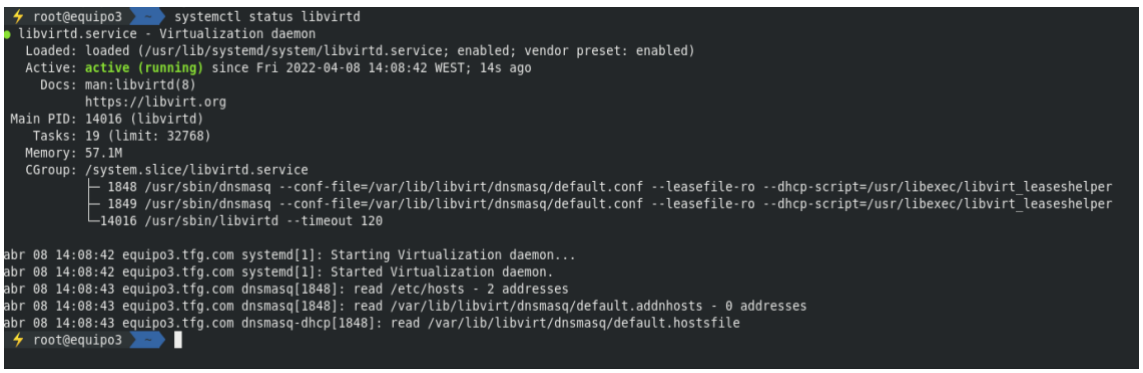

Ilustración 3: Verificación del estado del demonio libvirtd

En el caso de que el demonio "*libvirtd*" no esté activo, se procederá a iniciarlo y a habilitarlo en el arranque del sistema de la siguiente forma:

```
$ systemctl enable --now libvirtd
```

Nota: El flag *"--now"*, inicia el demonio además de que se habilite en el arranque del sistema.

13. Capítulo XI: Operaciones con máquinas virtuales

13.1. Creación de una máquina virtual mediante *virt-manager*

En este apartado, se procederá a crear una máquina virtual que ejecuta una distribución *Oracle Linux* haciendo uso de la herramienta gráfica de KVM *virt-manager*. Además, se realizarán pruebas para chequear el correcto funcionamiento de la máquina virtual. Los pasos para la creación de una máquina virtual mediante virt-manager se detallan en el Anexo IV [33, Cap. 3.1].

13.2. Copia de una máquina virtual

En esta sección, se procederá a realizar una copia de seguridad manual de la máquina virtual de KVM creada en el apartado anterior. Para ello y sabiendo que una máquina virtual no son más que ficheros, se realizará una copia de seguridad de los ficheros importantes de las máquinas virtuales en KVM, como pueden ser el fichero de configuración XML de la máquina virtual y la imagen del disco de la MV. Para ver los detalles de la realización de esta copia de seguridad manual, se puede revisar el Anexo V [34].

13.3. Clonación de una máquina virtual mediante virt-manager

En este apartado, se realizará una clonación completa (creación de una máquina virtual a partir de otra existente) de la máquina virtual creada e instalada como prueba en el apartado 11.1 del capítulo XI. Para ello, se hará uso de la herramienta gráfica proporcionada por KVM denominada "*virt-manager*" [33, Cap. 3.7]. Se puede ver más información detallada acerca de los pasos para llevar cabo la clonación en el Anexo VI.

13.4. Clonación de una máquina virtual mediante virt-clone

En este apartado, se realizará una clonación (creación de una máquina virtual a partir de otra existente) de la máquina virtual creada e instalada como prueba en el apartado 11.1 del capítulo XI llamada PruebaVirtManager. Para ello, se hará uso de la utilidad de línea de comandos "*virt-clone*" [35].

Para llevar a cabo esta clonación, ejecutamos en la *shell* de *bash* de Oracle Linux la siguiente orden, como se muestra en la ***Ilustración 4***:

```
root@equipo3 ~/Descargas virt-clone -o PruebaVirtManager --name Prueba_virt_clone -f /var/lib/libvirt/images/Prueba_virt_clone.qcow2 -m 56:31:1b:fa:1f:33
Asignando 'Prueba_virt_clone.qcow2'

El clon 'Prueba_virt_clone' ha sido creado exitosamente.
```

Ilustración 4: Clonación de una máquina virtual con la utilidad virt-clone

Donde en la opción "*-o*" de la utilidad de KVM "*virt-clone*" hace referencia al nombre de la máquina virtual original a clonar, la opción "--name" hace referencia al nuevo nombre de la máquina virtual clonada, la opción "-f" es la ruta dónde se va a guardar la imagen del disco de la máquina una vez clonada y la opción "-m" es la dirección MAC de la máquina virtual a clonar.

Nota: La nueva dirección MAC, la podemos generar haciendo uso del generador de direcciones MAC aleatorias usado anteriormente en el Anexo V.

Tras realizar los pasos anteriores, ya tendríamos nuestra nueva máquina virtual clonada con la utilidad de línea de comandos *virt-clone*. Al ponerla en funcionamiento debemos de hacer comprobaciones de red, configuraciones de IP (si fuera necesario), etc.

13.5. Creación e instalación de una nueva máquina virtual mediante virt-install

A continuación, se creará una nueva máquina virtual a través de la utilidad de línea de órdenes de KVM "*virt-install*" [32, Cap 3]. Las especificaciones de la nueva máquina virtual a crear son:

- Nombre: Prueba_virt_install
- RAM: 1GB
- CPU: 1
- Espacio en disco: 10GB
- S.O: Oracle Linux
- Red: NAT

Para la creación de esta nueva máquina virtual, ejecutamos en el shell de Oracle Linux la siguiente orden, como se muestra en la ***Ilustración 5***:

```
root@equipo3 ~/Descargas virt-install --name Prueba_virt_install --mem
ory 1024 --vcpus 1 --disk size=10 --location /root/Descargas/OracleLinux-R8-U5-x
86_64-dvd.iso --os-variant ol8.5 --network default
```

Ilustración 5: Creación de una nueva máquina virtual con virt-install

Nota: En la ***Ilustración 5***, se muestra el flag *"--os-variant*" con la versión 8.5 de Oracle Linux debido a que, en el momento de tomar esa foto, la versión 8.6 de Oracle Linux aún no estaba en funcionamiento.

Analizando las opciones del comando anteriormente ejecutado, la opción ***"--name***" de la utilidad de KVM "*virt-install*" hace referencia al nombre de la nueva máquina virtual a crear, la opción ***"--memory***" hace referencia a la memoria RAM que se le va a asignar a esta nueva máquina, la opción ***"--vcpus***" es el número de CPU'S que va tener la nueva máquina, la opción ***"--disk size***" es el tamaño de disco, ***"--cdrom***" hace referencia a la ISO, que es el sistema operativo a instalar en la nueva máquina virtual, ***"--os-variant***" que es la variante del S.O a instalar y por último ***"--network default***" que hace referencia a que la red de la nueva máquina virtual a crear va a ser NAT. Tras la ejecución del comando anterior, se le dará paso al instalador del S.O correspondiente.

Tras realizar la instalación y configuración de la nueva máquina virtual creada mediante la utilidad virt-install se deben hacer comprobaciones de red, configuraciones de IP (si fuera necesario), etc.

14. Capítulo XII: Recursos de almacenamiento virtual

Para desarrollar este capítulo y los anexos referenciados en él, se ha tenido en cuenta la documentación disponible en [35, Cap. 11], [37], [38].

14.1. Contenedor local: Creación de un volumen de 1GB en el contenedor Default y asociación a una máquina virtual

En este apartado, se creará un volumen con capacidad de 1GB en el contenedor por defecto (default) haciendo uso tanto de la herramienta gráfica "*virt-manager*" como de la utilidad de línea de comandos "*virsh*". Tras la creación del volumen, se asociará el mismo a una máquina virtual existente en KVM y se comprobará que se pueden almacenar datos en él. Para ver los detalles de todos los pasos para la creación y asociación del volumen se puede consultar el Anexo VII.

14.2. Contenedor local: Creación de una nueva partición en el host anfitrión y asociación a una máquina virtual

En este apartado, se creará una nueva partición lógica de 1GB sobre la partición extendida del disco físico "***sda***" del sistema anfitrión y se asociará dicha partición a una máquina virtual de KVM haciendo uso de la utilidad de línea de órdenes *virsh*. Además, se comprobará que se pueden almacenar datos en la partición recién creada desde la máquina virtual. El objetivo fundamental de este apartado es ofrecer a un sistema invitado la posibilidad de trabajar directamente sobre una partición del disco físico del sistema anfitrión. Para más detalle sobre cómo crear una partición en el sistema host y asociarla a una MV, ver Anexo VIII.

14.3. Contenedor local: Creación de un contenedor (*storage pool*) en una partición lógica del sistema anfitrión

En esta sección, se llevará a cabo la creación de un contenedor de almacenamiento (*storage pool*) de tipo "*Dispositivo de bloque preformateado (fs)"* en una partición lógica de 2GB del sistema anfitrión. Para ello, se deberá crear una nueva partición lógica de 2GB sobre la partición extendida del disco físico ***sda*** del host anfitrión (utilizando *fdisk*), formatearla en formato ext4 (*mkfs.ext4*) y hacer uso tanto de *virt-manager* como de *virsh* para la creación del contenedor de almacenamiento. Tras la creación del contenedor, se creará un nuevo volumen de 1GB llamado "*Vol3*" en el contenedor recién creado y se asociará a una máquina virtual. Por último, se comprobará que se pueden almacenar datos desde la MV en el nuevo volumen creado en el contenedor FS.

El objetivo fundamental de esta tarea es el crear un contenedor de almacenamiento donde poder crear volúmenes (discos virtuales) para los sistemas invitados. Para más detalle sobre los pasos de cómo realizar este procedimiento se puede consultar el Anexo IX, en dónde se crea la partición lógica de 2GB, el contenedor FS utilizando *virt-manager* y utilizando *virsh* y el

volumen de 1GB en el contenedor recién creado y su asociación a una máquina virtual haciendo uso de *virsh*.

14.4. Contenedor en red: Contenedor NFS de imágenes ISO

En este apartado se creará un nuevo contenedor de tipo *netfs*. Este contenedor en red alberga imágenes ISO de sistemas operativos, y para su creación se hará uso de un servicio NFS que proporciona imágenes ISO al host anfitrión. Este servicio NFS estará simulado, ya que se encuentra configurado en una máquina virtual de KVM. Para ver los detalles de la configuración del servidor NFS [39, Cap. 4] que exporta diferentes recursos al host anfitrión, se puede consultar el Anexo X.

Como se ha mencionado anteriormente, para la creación de este contenedor NFS se hará uso de la herramienta gráfica *virt-manager* y del servicio NFS simulado denominado "***servidor-oracle-linux***" y de su directorio "***/ISOS/OL/8.5***" que proporcionará una imagen ISO del sistema operativo Oracle Linux en su versión 8.5 al sistema anfitrión. Para ver en detalle los pasos para la creación de este contenedor en red se puede consular el Anexo XI.

14.5. Contenedor en red: Contenedor NFS de volúmenes de máquinas virtuales

En entornos de virtualización es muy habitual disponer de varios servidores virtuales que emplean un mismo servidor común de almacenamiento (cabina de almacenamiento) con el fin de proporcionar espacios a las máquinas virtuales [40]. Es por ello que en este apartado se simulará este escenario con la creación de un nuevo contenedor de tipo NFS en el sistema anfitrión que simule un contenedor de volúmenes de máquinas virtuales.

Se volverá a utilizar el servicio NFS simulado por la máquina "***servidor-oracle-linux***" y su directorio "***/volumenes***" que proporcionará volúmenes de máquinas virtuales. Para ver los detalles de la creación del contenedor NFS de volúmenes de máquinas virtuales haciendo uso de *virsh* y de la creación de un nuevo volumen de 1GB en él, se puede consultar el Anexo XII.

15. Capítulo XIII: Creación de un bridge en la interfaz física

Para este apartado y los anexos referenciados en él, se ha usado la documentación accesible en [38, Cap. 17], [41].

15.1. Creación del bridge

Libvirt utiliza el concepto de "*switch virtual*" que opera en el host anfitrión. Los *switchs virtuales* en KVM pueden operar en modo NAT (modo por defecto), en modo "Red Enrutada", en modo "Red Aislada" y en modo "Bridge". Cada vez que se crea una red en KVM, se crea automáticamente una interfaz con nombre "*virbrX"* en el host anfitrión, que hace referencia a un switch virtual. Además, cada interfaz de cada máquina virtual también es visible como una interfaz con nombre "*vnetX*" en el sistema anfitrión.

En este apartado, se procederá a crear un switch virtual asociado a la interfaz física del host anfitrión que opere en modo *bridge*. De esta manera, las máquinas virtuales que se conecten al bridge estarán en la misma red que la interfaz física del host anfitrión. Para ello y recordando que en el host anfitrión el servicio *NetworkManager* está deshabilitado, se deberá configurar un nuevo *script* de red para configurar el *bridge* (br0) e indicarle al *script* de red de la interfaz física que forma parte del *bridge*.

Una vez creado el bridge en la interfaz física del host anfitrión, se procederá a crear una nueva máquina virtual (con el servicio *NetworkManager* deshabilitado) y conectarla al bridge haciendo uso de *virt-manager*. Una vez arrancada la máquina virtual conectada al bridge, se deberá dar de alta la dirección MAC de la máquina virtual en el servidor DHCP del IUCTC para finalizar con la configuración y configurar su interfaz de red en modo DHCP. Para ver todos los detalles de la creación del bridge y la conexión de una máquina virtual al mismo se puede consultar el Anexo XIII.

15.2. Pruebas de conectividad del *bridge*

Una vez configurado el *bridge* en la interfaz física del host anfitrión y conectar una máquina virtual al mismo, se deberá comprobar que desde la máquina virtual se puede acceder al exterior y que desde el exterior se puede acceder a la MV (Ping desde una máquina externa) (Ver ***Ilustración 6***):

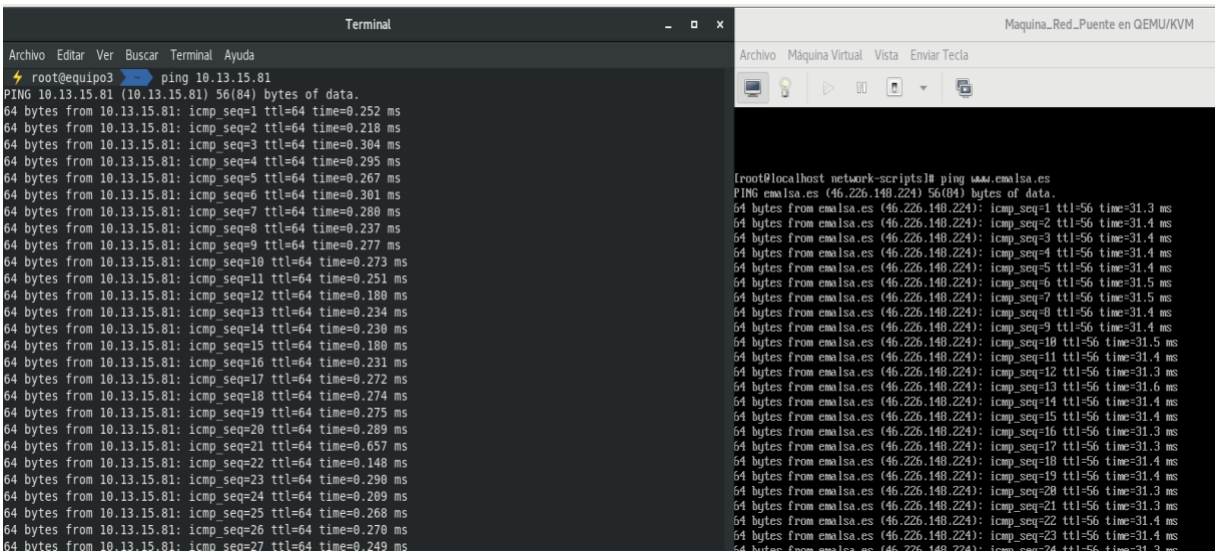

Ilustración 6: Pruebas de conectividad del bridge

16. Capítulo XIV: Diseño y despliegue de la Infraestructura de clúster en alta disponibilidad y con balanceo de carga

16.1. Descripción de la infraestructura del clúster

En este apartado se comenzará con el diseño de la infraestructura del clúster para obtener alta disponibilidad y balanceo de carga en los servicios a ofrecer por el mismo. El clúster estará conformado por cinco máquinas virtuales de KVM (nodos) y cada máquina ofrecerá un servicio en el funcionamiento final del clúster. Además, se dispondrá de una sexta máquina virtual que no formará parte del clúster, pero ofrecerá un almacenamiento compartido y distribuido *iSCSI* a los diferentes nodos del mismo.

Por otro lado, para lograr un buen funcionamiento de esta infraestructura de clúster, se necesita interconectar los diferentes nodos del clúster mediante una red de alta velocidad que minimice la latencia al mínimo en el proceso de intercambio de información entre los nodos del clúster. Para ello, se hará uso de tres redes virtuales: Red NAT (para permitir que los diferentes nodos tengan acceso al exterior y poder descargar el software necesario de los repositorios de internet), Red aislada de Almacenamiento (para controlar el tráfico de acceso al almacenamiento compartido) y Red aislada de Control (para tener un control del tráfico de la información entre los nodos que conforman el clúster). Esta segmentación de la red en tres redes distintas es deseable desde el punto de vista de la seguridad.

Acto seguido, se procederá a explicar la configuración de cada nodo que conforma la infraestructura y los servicios que ofrece cada uno:

- Dos nodos serán los encargados de la configuración del sistema de gestión de contenidos *Wordpress* y del propio servidor web *Apache* (ya que *Wordpress* está basado en Apache). Los nombres de las máquinas virtuales serán **Nodo1** y **Nodo2** y sus nombres de dominio completamente cualificados serán **nodo1.tfg.com** y **nodo2.tfg.com**, respectivamente.

- Otros dos nodos serán los encargados de proporcionar la replicación y la alta disponibilidad en el servicio de base de datos *MariaDB.* Para conseguir esta replicación y alta disponibilidad, se hará uso de *Galera Cluster*. Los nombres de las máquinas virtuales serán **Nodo3** y **Nodo4** y los nombres de dominio completamente cualificados serán **nodo3.tfg.com** y **nodo4.tfg.com**, respectivamente.

- Un nodo se encargará de proporcionar balanceo de carga y alta disponibilidad al servicio web ofrecido por el CMS *Wordpress* basado en Apache y también ofrecerá balanceo de carga en las peticiones que se realicen a la base de datos MariaDB de los nodos 3 y 4. Para todo ello, se configurará el servicio *HAProxy*. El nombre de esta máquina virtual será **Balanceador-De-Carga** y su nombre de dominio completamente cualificado es **balanceador.tfg.com**.

- Un nodo será el encargado de proporcionar un almacenamiento compartido y distribuido a los diferentes nodos del clúster. Para ello, se configurará en él, el servicio *iSCSI* como servidor. Este nodo no formará parte del clúster. El nombre de esta máquina virtual será **Almaceamiento-iSCSI** y su nombre de dominio completamente cualificado es **iscsi.tfg.com**.

En la figura siguiente se puede observar la representación del diseño de la infraestructura del clúster que se ha descrito anteriormente (Ver ***Ilustración 7***):

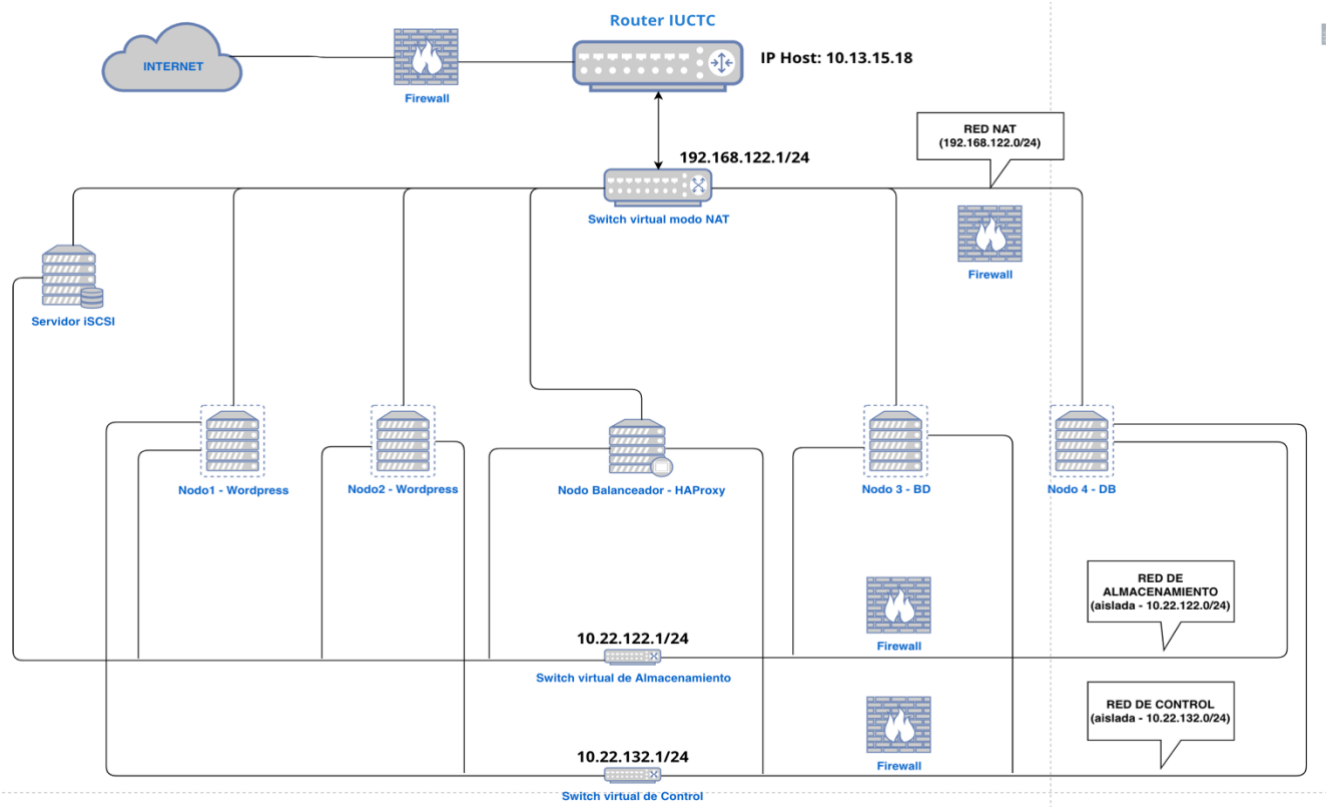

Ilustración 7: Diagrama representativo de la infraestructura del clúster

Analizando con más profundidad la figura anterior, se puede observar que se dispone de seis servidores (máquinas virtuales), dónde cinco de ellos hacen referencia a los nodos del clúster (Nodo1, Nodo2, Nodo3, Nodo4 y Balanceador) y el otro servidor hace referencia a la máquina virtual que proporciona un almacenamiento compartido iSCSI a los nodos del clúster. Por otro lado, se dispone de tres *switchs* virtuales, dónde dos de ellos (Switch virtual de Control y Switch virtual de Almacenamiento) operan en modo aislado y el otro switch denominado “Switch virtual modo NAT” opera en modo NAT. El switch virtual de almacenamiento hace referencia a la propia “Red de Almacenamiento (10.22.122.0/24)”, el switch de control hace referencia a la propia “Red de Control (10.22.132.0/24)” y el switch virtual que opera en modo NAT hace referencia a la

"Red default" de tipo NAT. Como se puede observar, a los *switchs* virtuales de Almacenamiento y de NAT, se conectan todas las máquinas virtuales, mientras que al switch virtual de Control sólo se conectan los nodos del clúster (Nodo1, Nodo2, Nodo3, Nodo4 y Balanceador). Los *switchs* virtuales en modo aislado tan solo permiten la conexión con los nodos que están en su red y con el host anfitrión, mientras que switch virtual que opera en modo NAT permite el acceso al exterior mediante el *router* del IUCTC (ya que este trabajo se ha realizado en el IUCTC). Además, el tráfico en cada máquina virtual (nodos) y en el sistema anfitrión, en cada interfaz de red, está controlado por un cortafuegos (firewalld).

Por último, mencionar que en cada nodo de la infraestructura se hará uso del servicio *NetworkManager* para la configuración de red y del servicio de seguridad SELinux en modo "*enforcing*" la mayor parte del tiempo (ya que en configuraciones posteriores, para disponer de un almacenamiento compartido síncrono haciendo uso del sistema de archivos OCFS2, se debe deshabilitar).

16.2. Infraestructura de red del clúster

Como ya se ha comentado anteriormente, para la comunicación entre las diferentes máquinas virtuales del clúster se harán uso de tres redes virtuales: Una Red NAT y dos redes aisladas.

16.2.1. Red NAT

Esta red virtual ofrecerá a los diferentes nodos que conforman la infraestructura, la posibilidad de acceder al exterior y poder descargar el software necesario desde los repositorios que está en internet y, además, permitirá el tráfico de los servicios del clúster al exterior. Esta red recibirá el nombre de "**default**" y no hará falta crearla, ya que tras la instalación de la plataforma KVM se crea por defecto y es visible en el sistema anfitrión como una interfaz de red denominada **virbr0**. La dirección de red será la "**192.168.122.0/24**" y se hará uso

del servicio DHCP. Esta red será usada por todos los nodos de la infraestructura (Nodo1, Nodo2, Nodo3, Nodo4, Balanceador-De-Carga y Almacenamiento-iSCSI). En la ***Ilustración 8*** se puede observar esta red que opera en modo NAT:

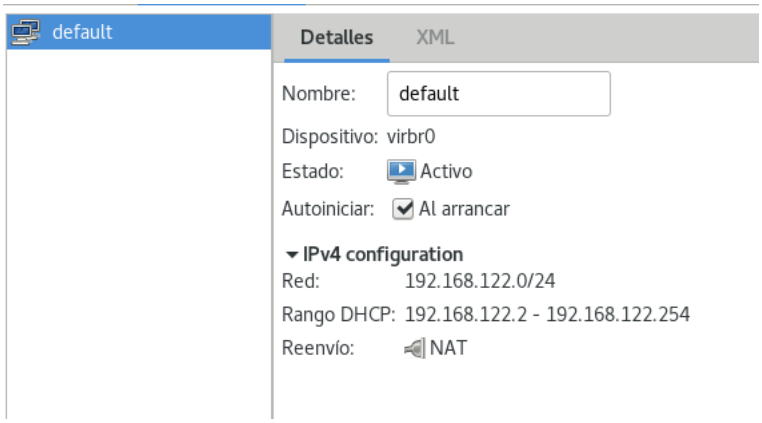

Ilustración 8: Red virtual de tipo NAT del clúster

16.2.2. Red aislada de Almacenamiento

Esta red virtual recibirá el nombre de "**Almacenamiento**" y operará en modo aislada (*isolated*), es decir, los nodos conectados a esta red tendrán la posibilidad de comunicarse entre sí y con el sistema anfitrión, pero no tendrán acceso al exterior ni tendrán la posibilidad de recibir solicitudes desde fuera del host anfitrión. Esta red tendrá la función de controlar el tráfico de acceso al almacenamiento compartido proporcionado por el servidor externo *iSCSI*. La dirección de red será **"10.22.122.0/24"** y no se hará uso del servicio DHCP, ya que las direcciones de cada nodo en esta red se establecerán de manera estática. Esta red será usada por todos los nodos de la infraestructura (Nodo1, Nodo2, Nodo3, Nodo4, Balanceador-De-Carga y Almacenamiento-iSCSI). Para ver los detalles de la creación de esta red se puede consultar el Anexo XIV.

16.2.3. Red aislada de Control

Esta red virtual recibirá el nombre de "**Control**" y operará en modo aislada (*isolated*), teniendo la posibilidad de comunicarse con otras máquinas virtuales

que estén en su red y con el sistema anfitrión, pero no permitirá el acceso al exterior ni tendrá la posibilidad de recibir solicitudes desde fuera del host anfitrión. Esta red tendrá la función de controlar el tráfico que circula en la comunicación entre los diferentes nodos que conforman el clúster. La dirección de red será **"10.22.132.0/24"** y no se hará uso del servicio DHCP, ya que las direcciones de cada nodo en esta red se establecerán de manera estática. Esta red será usada únicamente por los nodos del clúster (Nodo1, Nodo2, Nodo3, Nodo4 y Balanceador-De-Carga), ya que el nodo "Almacenamiento-iSCSI" no conforma parte del clúster y por tanto no la usará. Para ver los detalles de la creación de esta red se puede consultar el Anexo XV.

16.3. Creación de la máquina virtual base

Una vez diseñada la infraestructura básica que tendrá el clúster, se procederá con la creación de las diferentes máquinas virtuales (nodos) que conforman la infraestructura mediante *virt-manager*. Para facilitar este trabajo, se creará una "**Máquina Virtual Base**" con una única interfaz de red conectada a la Red NAT (default) y con las siguientes características:

- 1 CPU
- 1 GB de RAM
- 1 disco de 15GB
- Instalación **mínima** de Oracle Linux 8.6

Para ver en más detalle los pasos para la creación de una máquina virtual en KVM mediante *virt-manager* se pueden consultar el Anexo IV.

Una vez creada la máquina virtual base, tomará el control el instalador del sistema operativo. Tras la instalación, se procederá a actualizar el sistema e instalar algunas utilidades que pueden ser bastante útiles en el transcurso de la configuración.

Primero que nada, se actualizará el sistema de la máquina virtual base con el siguiente comando:

```
$ dnf upgrade
```

Y acto seguido, se procederá con la instalación de algunas utilidades básicas de red, como puede ser el comando "*ifconfig*", con la siguiente instrucción:

```
$ dnf -y install net-tools
```

Por último, para concluir con la configuración de la "Máquina virtual Base" se reiniciará la misma:

```
$ reboot
```

16.4. Creación de los diferentes nodos de la infraestructura

Una vez se disponga de la "Máquina Virtual Base" creada, instalada y actualizada, se procederá con la creación de los seis nodos que conformaran la infraestructura final, clonando la "Máquina virtual Base" mediante *virt-manager*. Para ver los detalles de la clonación de una máquina virtual en KVM mediante *virt-manager* se puede consultar el Anexo VI.

Una vez clonada la "Máquina virtual Base" seis veces (número de nodos de la infraestructura), se tendrán todas las máquinas virtuales necesarias para el diseño y despliegue del clúster en alta disponibilidad y con balanceo de carga (Ver ***Ilustración 9***).

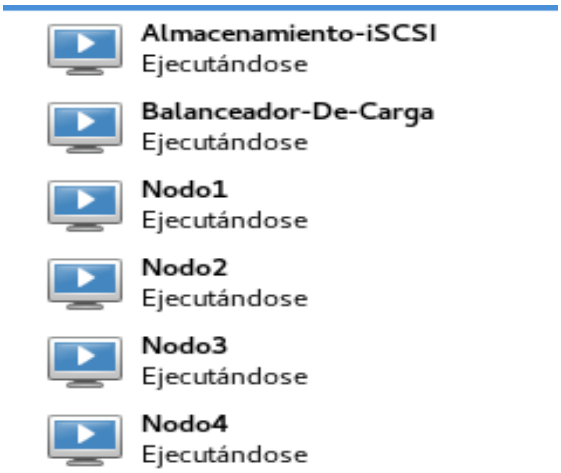

Ilustración 9: Nodos de la infraestructura del TFG

Nota: Cada máquina virtual dispondrá de una única interfaz de red en modo NAT, debido a que la "Máquina Virtual Base" se creó con esa única interfaz de red. Además, es muy recomendable probar que se tiene conectividad hacia el exterior en cada nodo.

Por último, se deberá añadir un nuevo disco de almacenamiento con una capacidad de 10GB a la máquina virtual "**Almacenamiento iSCSI**". Este disco será añadido con el fin de poder ser ofrecido como almacenamiento compartido a los diferentes nodos del clúster. En el Anexo XVI se puede consultar los detalles para añadir un nuevo disco de almacenamiento al nodo "**Almacenamiento iSCSI**".

16.5. Creación de las interfaces de red de cada nodo

En este apartado, se procederá a asignar y configurar las distintas interfaces de red de cada máquina virtual. Cabe recordar que cada máquina virtual ya dispone de una única interfaz de red asignada (Red default en modo NAT) y que no hay que configurarla, debido a que el servicio DHCP se encarga de ello. Pero también cabe recordar, que los diferentes nodos usarán para su comunicación otras redes (Red de Control y de Almacenamiento). Por ello, en

primer lugar, se deberán asignar las diferentes redes a cada máquina virtual mediante el panel de hardware de *virt-manager*. Este proceso es muy similar al descrito en el Anexo XVI, pero en lugar de agregar un dispositivo de almacenamiento, se deberá agregar un dispositivo de red y seleccionar la red a agregar y su modelo de dispositivo (*virtio*).

En todos los nodos del clúster (Nodo1, Nodo2, Nodo3, Nodo4 y Balanceador-De-Carga), se agregarán como dispositivos de red mediante el panel de hardware de *virt-manager,* mencionado anteriormente, las redes aisladas de Almacenamiento y de Control. Mientras que en el nodo "Almacenamiento-iSCSI" (que no forma parte del clúster) tan solo se añadirá como dispositivo de red la red aislada de "Almacenamiento".

Una vez agregados todos los dispositivos de red en los diferentes nodos como se ha comentado en el párrafo anterior, se procederá con la configuración de cada interfaz de red en cada máquina virtual haciendo uso de "***NetworkManager***" y su utilidad "***nmcli***" [42].

16.5.1 Configuración de las interfaces de red del Nodo1

Con respecto al **Nodo1**, se tendrán tres interfaces de red. La primera interfaz (**enp1s0**), permitirá la conexión con la red **NAT** por defecto (default) y se tendrá como dirección IP la **192.168.122.137/24** proporcionada por el servidor DHCP. Esta interfaz proporcionará conexión al exterior al nodo. La segunda interfaz de red (**enp7s0**), permitirá la conexión con la red aislada de "**Almacenamiento**" con la dirección **10.22.122.10/24**. Por esta segunda interfaz se controlará el acceso al almacenamiento compartido por parte del nodo. Por último, se tendrá una tercera interfaz de red (**enp8s0**), que permitirá la conexión con la red aislada de "**Control**" mediante la dirección **10.22.132.10/24**. Esta interfaz permitirá al nodo tener un control en el tráfico de información entre los nodos del clúster.

En primer lugar, se comenzará configurando la interfaz enp7s0 correspondiente a la red aislada de Almacenamiento. Para ello, se añadirá mediante la utilidad "*nmcli*" una nueva conexión llamada "**red-almacenamiento**" y se configurará de manera estática su dirección IP:

```
$ nmcli con add con-name red-almacenamiento ipv4.method manual
ifname enp7s0 type ethernet ipv4.addresses '10.22.122.10/24'
connection.autoconnect yes
```

Acto seguido, se deberá activar la interfaz recién configurada de la siguiente manera:

```
$ nmcli device connect enp7s0
```

En segundo lugar, se configurará la interfaz enp8s0 correspondiente a la red aislada de Control. Para ello, se añadirá mediante la utilidad "*nmcli*" una nueva conexión llamada "**red-control**" y se configurará de manera estática su dirección IP:

```
$ nmcli con add con-name red-control ipv4.method manual
ifname enp8s0 type ethernet ipv4.addresses '10.22.132.10/24'
connection.autoconnect yes
```

Acto seguido, se deberá activar la interfaz recién configurada de la siguiente manera:

```
$ nmcli device connect enp8s0
```

Por último, es muy recomendable analizar el estado de las conexiones establecidas haciendo uso del comando "***nmcli device status***" y comprobar la conectividad por cada interfaz haciendo uso del comando "*ping*". Para comprobar la conectividad de la interfaz de la red aislada de almacenamiento, se puede hacer un ping a su puerta de enlace (**ping -I enp7s0 10.22.122.1**) o un ping a

una de las máquinas de su red. Para comprobar la conectividad de la interfaz de la red aislada de control, se puede hacer un ping a su puerta de enlace (**ping -I enp8s0 10.22.132.1**) o un ping a una de las máquinas de su red. Mientras que para comprobar la conectividad por la interfaz de la red NAT se puede ejecutar: **ping -I enp1s0 www.google.es**. Se pueden ver los detalles de las interfaces de red del Nodo1 en el Anexo XVII.

16.5.2 Configuración de las interfaces de red del Nodo2

Con respecto al **Nodo2**, se tendrán tres interfaces de red. La primera interfaz (**enp1s0**), permitirá la conexión con la red **NAT** por defecto (default) y se tendrá como dirección IP la **192.168.122.124/24** proporcionada por el servidor DHCP. Esta interfaz proporcionará conexión al exterior al nodo. La segunda interfaz de red (**enp7s0**), permitirá la conexión con la red aislada de

"**Almacenamiento**" con la dirección **10.22.122.11/24**. Por esta segunda interfaz se controlará el acceso al almacenamiento compartido por parte del nodo. Por último, se tendrá una tercera interfaz de red (**enp8s0**), que permitirá la conexión con la red aislada de "**Control**" mediante la dirección **10.22.132.11/24**. Esta interfaz permitirá al nodo tener un control en el tráfico de información entre los nodos del clúster.

En primer lugar, se comenzará configurando la interfaz enp7s0 correspondiente a la red aislada de Almacenamiento. Para ello, se añadirá mediante la utilidad "*nmcli*" una nueva conexión llamada "**red-almacenamiento**" y se configurará de manera estática su dirección IP:

```
$ nmcli con add con-name red-almacenamiento ipv4.method manual
ifname enp7s0 type ethernet ipv4.addresses '10.22.122.11/24'
connection.autoconnect yes
```

Acto seguido, se deberá activar la interfaz recién configurada de la siguiente manera:

```
$ nmcli device connect enp7s0
```

En segundo lugar, se configurará la interfaz enp8s0 correspondiente a la red aislada de Control. Para ello, se añadirá mediante la utilidad "*nmcli*" una nueva conexión llamada "**red-control**" y se configurará de manera estática su dirección IP:

```
$ nmcli con add con-name red-control ipv4.method manual
ifname enp8s0 type ethernet ipv4.addresses '10.22.132.11/24'
connection.autoconnect yes
```

Acto seguido, se deberá activar la interfaz recién configurada de la siguiente manera:

```
$ nmcli device connect enp8s0
```

Por último, es muy recomendable realizar las mismas comprobaciones que se llevaron a cabo en la configuración de las interfaces de red del Nodo1. Se pueden ver los detalles de las interfaces de red del Nodo2 en el Anexo XVIII.

16.5.3 Configuración de las interfaces de red del Nodo3

Con respecto al **Nodo3**, se tendrán tres interfaces de red. La primera interfaz (**enp1s0**), permitirá la conexión con la red **NAT** por defecto (default) y se tendrá como dirección IP la **192.168.122.199/24** proporcionada por el servidor DHCP. Esta interfaz proporcionará conexión al exterior al nodo. La segunda interfaz de red (**enp7s0**), permitirá la conexión con la red aislada de "**Almacenamiento**" con la dirección **10.22.122.12/24**. Por esta segunda interfaz

se controlará el acceso al almacenamiento compartido por parte del nodo. Por último, se tendrá una tercera interfaz de red (**enp8s0**), que permitirá la conexión con la red aislada de "**Control**" mediante la dirección **10.22.132.12/24**. Esta interfaz permitirá al nodo tener un control en el tráfico de información entre los nodos del clúster.

En primer lugar, se comenzará configurando la interfaz enp7s0 correspondiente a la red aislada de Almacenamiento. Para ello, se añadirá mediante la utilidad "*nmcli*" una nueva conexión llamada "**red-almacenamiento**" y se configurará de manera estática su dirección IP:

```
$ nmcli con add con-name red-almacenamiento ipv4.method manual
ifname enp7s0 type ethernet ipv4.addresses '10.22.122.12/24'
connection.autoconnect yes
```

Acto seguido, se deberá activar la interfaz recién configurada de la siguiente manera:

```
$ nmcli device connect enp7s0
```

En segundo lugar, se configurará la interfaz enp8s0 correspondiente a la red aislada de Control. Para ello, se añadirá mediante la utilidad "*nmcli*" una nueva conexión llamada "**red-control**" y se configurará de manera estática su dirección IP:

```
$ nmcli con add con-name red-control ipv4.method manual
ifname enp8s0 type ethernet ipv4.addresses '10.22.132.12/24'
connection.autoconnect yes
```

Acto seguido, se deberá activar la interfaz recién configurada de la siguiente manera:

```
$ nmcli device connect enp8s0
```

Por último, es muy recomendable realizar las mismas comprobaciones que se llevaron a cabo en la configuración de las interfaces de red del Nodo1. Se pueden ver los detalles de las interfaces de red del Nodo3 en el Anexo XIX.

16.5.4 Configuración de las interfaces de red del Nodo4

Con respecto al **Nodo4**, también se tendrán tres interfaces de red. La primera interfaz (**enp1s0**), permitirá la conexión con la red **NAT** por defecto (default) y se tendrá como dirección IP la **192.168.122.219/24** proporcionada por el servidor DHCP. La segunda interfaz de red (**enp7s0**), permitirá la conexión con la red aislada de "**Almacenamiento**" con la dirección **10.22.122.13/24**. Por último, se tendrá una tercera interfaz de red (**enp8s0**), que permitirá la conexión con la red aislada de "**Control**" mediante la dirección **10.22.132.13/24**.

En primer lugar, se comenzará configurando la interfaz enp7s0 correspondiente a la red aislada de Almacenamiento. Para ello, se añadirá mediante la utilidad "*nmcli*" una nueva conexión llamada "**red-almacenamiento**" y se configurará de manera estática su dirección IP:

```
$ nmcli con add con-name red-almacenamiento ipv4.method manual
ifname enp7s0 type ethernet ipv4.addresses '10.22.122.13/24'
connection.autoconnect yes
```

Acto seguido, se deberá activar la interfaz recién configurada de la siguiente manera:

```
$ nmcli device connect enp7s0
```

En segundo lugar, se configurará la interfaz enp8s0 correspondiente a la red aislada de Control. Para ello, se añadirá mediante la utilidad "*nmcli*" una

nueva conexión llamada "**red-control**" y se configurará de manera estática su dirección IP:

```
$ nmcli con add con-name red-control ipv4.method manual
ifname enp8s0 type ethernet ipv4.addresses '10.22.132.13/24'
connection.autoconnect yes
```

Acto seguido, se deberá activar la interfaz recién configurada de la siguiente manera:

```
$ nmcli device connect enp8s0
```

Por último, es muy recomendable realizar las mismas comprobaciones que se llevaron a cabo en la configuración de las interfaces de red del Nodo1. Se pueden ver los detalles de las interfaces de red del Nodo4 en el Anexo XX.

16.5.5 Configuración de las interfaces de red del Balanceador-De-Carga

Con respecto al **Balanceador-De-Carga**, también se tendrán tres interfaces de red. La primera interfaz (**enp1s0**), permitirá la conexión con la red **NAT** por defecto (default) y se tendrá como dirección IP la **192.168.122.209/24** proporcionada por el servidor DHCP. La segunda interfaz de red (**enp7s0**), permitirá la conexión con la red aislada de "**Almacenamiento**" con la dirección **10.22.122.14/24**. Por último, se tendrá una tercera interfaz de red (**enp8s0**), que permitirá la conexión con la red aislada de "**Control**" mediante la dirección **10.22.132.14/24**.

En primer lugar, se comenzará configurando la interfaz enp7s0 correspondiente a la red aislada de Almacenamiento. Para ello, se añadirá mediante la utilidad "*nmcli*" una nueva conexión llamada "**red-almacenamiento**" y se configurará de manera estática su dirección IP:

```
$ nmcli con add con-name red-almacenamiento ipv4.method manual
ifname enp7s0 type ethernet ipv4.addresses ‘10.22.122.14/24’
connection.autoconnect yes
```

Acto seguido, se deberá activar la interfaz recién configurada de la siguiente manera:

```
$ nmcli device connect enp7s0
```

En segundo lugar, se configurará la interfaz enp8s0 correspondiente a la red aislada de Control. Para ello, se añadirá mediante la utilidad “*nmcli*” una nueva conexión llamada “**red-control**” y se configurará de manera estática su dirección IP:

```
$ nmcli con add con-name red-control ipv4.method manual
ifname enp8s0 type ethernet ipv4.addresses ‘10.22.132.14/24’
connection.autoconnect yes
```

Acto seguido, se deberá activar la interfaz recién configurada de la siguiente manera:

```
$ nmcli device connect enp8s0
```

Por último, es muy recomendable realizar las mismas comprobaciones que se llevaron a cabo en la configuración de las interfaces de red del Nodo1. Se pueden ver los detalles de las interfaces de red del Balanceador-De-Carga en el Anexo XXI.

16.5.6 Configuración de las interfaces de red del Almacenamiento-iSCSI

Con respecto al nodo **Almacenamiento-iSCSI**, se tendrán dos interfaces de red. La primera interfaz (**enp1s0**), permitirá la conexión con la red **NAT** por defecto (default) y se tendrá como dirección IP la **192.168.122.101/24** proporcionada por el servidor DHCP. La segunda interfaz de red (**enp9s0**),

permitirá la conexión con la red aislada de "**Almacenamiento**" con la dirección **10.22.122.15/24**.

Por lo tanto, se comenzará configurando la interfaz enp9s0 correspondiente a la red aislada de Almacenamiento. Para ello, se añadirá mediante la utilidad "*nmcli*" una nueva conexión llamada "**red-almacenamiento**" y se configurará de manera estática su dirección IP:

```
$ nmcli con add con-name red-almacenamiento ipv4.method manual
ifname enp9s0 type ethernet ipv4.addresses '10.22.122.15/24'
connection.autoconnect yes
```

Acto seguido, se deberá activar la interfaz recién configurada de la siguiente manera:

```
$ nmcli device connect enp9s0
```

Por último, es muy recomendable realizar las mismas comprobaciones que se llevaron a cabo en la configuración de las interfaces de red del Nodo1, excepto por la interfaz de la red de control, ya que esta máquina carece de esta interfaz. Se pueden ver los detalles de las interfaces de red del Almacenamiento-iSCSI en el Anexo XXII.

16.6. Configuración del nombre de dominio de cada nodo

Tras la asignación de las diferentes interfaces de red en cada nodo y comprobar que la conectividad por cada una de ellas es la correcta, se procederá a establecer el nombre de dominio completamente cualificado en cada máquina virtual correspondiente:

Para establecer el nombre de dominio del Nodo1 se ejecutará:

```
$ hostnamectl set-hostname nodo1.tfg.com
```

Para establecer el nombre de dominio del Nodo2 se ejecutará:

```
$ hostnamectl set-hostname nodo2.tfg.com
```

Para establecer el nombre de dominio del Nodo3 se ejecutará:

```
$ hostnamectl set-hostname nodo3.tfg.com
```

Para establecer el nombre de dominio del Nodo4 se ejecutará:

```
$ hostnamectl set-hostname nodo4.tfg.com
```

Para establecer el nombre de dominio del Balanceador-De-Carga se ejecutará:

```
$ hostnamectl set-hostname balanceador.tfg.com
```

Para establecer el nombre de dominio del Almacenamiento-iSCSI se ejecutará:

```
$ hostnamectl set-hostname iscsi.tfg.com
```

Tras modificar el nombre de dominio en cada máquina virtual, se deberá reiniciar (reboot) cada nodo para que se apliquen los cambios.

16.7. Configuración del fichero hosts en cada nodo

A continuación, se procederá a añadir los nombres de las diferentes máquinas que intervienen en la infraestructura del clúster y sus direcciones IP en el fichero "***/etc/hosts***" para permitir la comunicación entre los diferentes nodos mediante los nombres de dominio.

El contenido del fichero ***"/etc/hosts***" de los nodos nodo1.tfg.com, nodo2.tfg.com, nodo3.tfg.com, nodo4.tfg.com y balanaceador.tfg.com es:

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.22.132.10 nodo1.tfg.com
10.22.132.11 nodo2.tfg.com
10.22.132.12 nodo3.tfg.com
10.22.132.13 nodo4.tfg.com
10.22.132.14 balanceador.tfg.com
10.22.122.15 iscsi.tfg.com
```

Ilustración 10: Fichero hosts de los nodos que conforman el clúster

El contenido del fichero ***"/etc/hosts***" del nodo iscsi.tfg.com es:

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.22.122.10 nodo1.tfg.com
10.22.122.11 nodo2.tfg.com
10.22.122.12 nodo3.tfg.com
10.22.122.13 nodo4.tfg.com
10.22.122.14 balanceador.tfg.com
10.22.122.15 iscsi.tfg.com
```

Ilustración 11: Fichero hosts del nodo del servidor iSCSI

Por último, es conveniente comprobar la conectividad (con la orden "*ping*") en cada máquina virtual usando los nombres de dominio completamente cualificados.

16.8. Instalación del servidor web *Apache*

Tras finalizar el diseño de toda la infraestructura básica del clúster, se procederá a instalar el servidor en el que se basará el servicio web ofrecido por *Wordpress [43]*. Este servidor web recibe el nombre de *Apache* y como se ha mencionado anteriormente, será configurado en los nodos nodo1.tfg.com y nodo2.tfg.com. Por lo tanto, tanto en el Nodo1 como en el Nodo2, se deberá instalar Apache de la siguiente manera:

```
$ dnf -y install httpd
```

Por otro lado, para permitir que Apache pueda proporcionar solicitudes a través de HTTP y HTTPS, se deberán añadir estos dos servicios al cortafuegos del Nodo1 y del Nodo2:

```
$ firewall-cmd --permanent --add-service http
```

```
$ firewall-cmd --permanent --add-service https
```

```
$ firewall-cmd --reload
```

Además, se debe iniciar y habilitar en el arranque del sistema el servicio Apache (*httpd*), Para ello:

```
$ systemctl enable --now httpd
```

Por último, se debe comprobar que el servicio Apache se ha iniciado sin errores (*systemctl status httpd*) y verificar desde el navegador del host anfitrión que el servidor web Apache se muestra correctamente tanto en el Nodo1 como en el Nodo2. Las pruebas del correcto funcionamiento de Apache se pueden consultar en el Anexo XXIII.

16.9. Instalación y configuración del servicio *HAProxy* para ofrecer alta disponibilidad y balanceo de carga al servidor web Apache

En esta sección, se instalará y se configurará en el nodo **Balanceador-De-Carga** el servicio *HAProxy*. Este servicio es un proxy inverso que proporcionará alta disponibilidad y balanceo de carga al servidor web Apache, en el cual está basado el servicio *Wordpress* a ofrecer. Además, este equilibrador

de carga distribuirá las peticiones que realicen los clientes del servicio web mediante un algoritmo de equilibrio de carga (en este caso Round Robin).

Para la instalación de *HAProxy* se ha consultado el manual oficial de Oracle que habla sobre ello [44]. Por lo tanto, en primer lugar, se instalará el servicio *HAProxy* en el nodo balanceador.tfg.com:

```
$ dnf -y install haproxy
```

Acto seguido, se deberá modificar el fichero de configuración de *HAProxy* para proporcionar alta disponibilidad y balanceo de carga al servidor web Apache [45]. Para ello, se tendrá que agregar al final del fichero ***"/etc/haproxy/haproxy.cfg***" el siguiente contenido (Ver ***Ilustración 12***):

```
# Frontend node that offers the Apache service with load balancing
frontend http_web
    bind 192.168.122.209:80
    mode http
    stats enable
    default_backend httpd_service

# Backend servers that offers the Apache Server
backend httpd_service
    balance roundrobin
    mode http
    server nodo1 10.22.132.10:80 check
    server nodo2 10.22.132.11:80 check
```

Ilustración 12: Secciones del fichero de configuración de HAProxy para ofrecer alta disponibilidad y balanceo de carga al servidor Apache

Analizando la figura anterior, se puede observar que se han añadido dos secciones al fichero de configuración de *HAProxy*: una sección **frontend** llamada **"***http_web***"** que escucha en el puerto 80, dirigiendo el tráfico al valor predeterminado de la sección backend denominada "*httpd_service*". Esta sección frontend define los puertos que aceptan conexiones de clientes y además proporciona una dirección IP (**192.168.122.209**, dirección IP de la interfaz enp1s0 del nodo balanceador.tfg.com) para acceder al servicio web. Por otro lado, en la sección **backend,** el algoritmo de equilibrio de carga establecido

para distribuir las peticiones es el de ***roundrobin***. Además, en la sección backend se definen los servidores (10.22.132.10 y 10.22.132.11, direcciones IP de control de los nodos nodo1.tfg.com y nodo2.tfg.com) a los que el proxy reenvía las conexiones de los clientes mediante la dirección IP cliente 192.168.122.209, definida en la sección frontend.

A continuación, es necesario habilitar en el cortafuegos los servicios que manejará *HAProxy*. En este caso se añadirá al firewall el servicio "http":

```
$ firewall-cmd --permanent --add-service http
```

```
$ firewall-cmd --reload
```

Tras añadir los servicios necesarios al firewall, se deberá iniciar y habilitar e el arranque del sistema el servicio *haproxy*. Para ello se debe ejecutar en el nodo Balanceador-De-Carga:

```
$ systemctl enable haproxy
```

```
$ systemctl start haproxy
```

Por último, es muy conveniente modificar la página inicial de Apache para saber que nodo (Nodo1 o Nodo2) son los que están ofreciendo el servicio en cada momento y observar cómo se distribuyen las peticiones entre ambos. Para ello, se deberá crear el fichero *"/var/www/html/index.html*" en los nodos 1 y 2 y agregar en él: **"<h1>Soy el Nodo1</h1>",** en el fichero del Nodo1 y **"<h1>Soy el Nodo2</h1>**" en el fichero del Nodo2.

Nota: Cabe recordar que el servicio Apache es ofrecido por los nodos Nodo1 y Nodo2 y se ha configurado *HAProxy* para que las peticiones a este servicio se distribuyan equitativamente.

Tras la realización de todos los pasos anteriores, se tendrá acceso al servidor Apache con alta disponibilidad y balanceo de carga desde el navegador del host anfitrión, haciendo uso de la dirección cliente configurada en el servicio *HAProxy*, 192.168.122.209. En el Anexo XXIV se pueden consultar las comprobaciones que justifican el equilibrio de carga y la alta disponibilidad del servidor web *Apache*.

16.10. Instalación y configuración del software High Availability Add-On

Una vez que se disponga de balanceo de carga y alta disponibilidad en el servidor web Apache de los Nodos 1 y 2, se comenzará a configurar el clúster que estará conformado por los nodos: **nodo1.tfg.com**, **nodo2.tfg.com**, **nodo3.tfg.com**, **nodo4.tfg.com** y **balanceador.tfg.com**. Para ello, se instalará el soporte parala computación en clúster (*High Availability Add-On*) [46] en los nodos 1, 2, 3, 4 y en el Balanceador-De-Carga.

En primer lugar, se deben habilitar los repositorios software de *Oracle Linux* que dan acceso al software de computación clúster. Este paso se debe de realizar en todos los nodos del clúster:

```
$ dnf config-manager --enable ol8_appstream ol8_baseos_latest
ol8_addons
```

Acto seguido, se deben instalar los paquetes software para el correcto funcionamiento del clúster (**pcs y pacemaker**) junto con los agentes disponibles del canal de alta disponibilidad y los agentes de recursos disponibles. Para ello, en cada nodo del clúster se debe ejecutar:

```
$ dnf -y install pcs pacemaker resource-agents fence-agents-all
```

Además, como se está ejecutando "*firewalld*", se debe agregar el servicio "*high-availability*" al cortafuegos de cada nodo del clúster para que los componentes del servicio se puedan comunicar a través de la red:

```
$ firewall-cmd --permanent --add-service high-availability
```

```
$ firewall-cmd --reload
```

Para usar el comando "***pcs***" para administrar y configurar el clúster, se debe establecer una contraseña al usuario "***hacluster***" que ha sido creado durante la instalación. Esta contraseña se debe establecer en cada nodo del clúster (nodo1.tfg.com, nodo2.tfg.com, nodo3.tfg.com, nodo4.tfg.com y balanceador.tfg.com). Se recomienda que la contraseña sea la misma en cada nodo. Para establecer la contraseña se hará uso de la utilidad "*passwd*":

```
$ passwd hacluster
```

Hay que tener en cuenta que para poder utilizar la orden "***pcs***" se debe tener activo y habilitado en cada nodo del clúster el servicio "***pcsd***". Por lo que en cada del clúster se debe ejecutar los comandos:

```
$ systemctl enable pcsd.service
```

```
$ systemctl start pcsd.service
```

Una vez que se tenga instalado todo el software necesario y sus configuraciones pertinentes, se comenzará con la creación del clúster. Lo primero que se debe hacer, es autenticar al usuario "***hacluster***" en los propios nodos del clúster. Esto se llevará a cabo mediante la utilidad "**pcs**" y su opción

"**auth**". Esta autenticación no hace falta que se lleve a cabo en todos los nodos del clúster, ya que es un comando que hace referencia a la administración del propio clúster, por lo que la autenticación del usuario "*hacluster*" se llevará a cabo en tan solo un nodo. En este caso, se hará uso del Nodo1:

```
$ pcs host auth nodo1.tfg.com addr=10.22.132.10 nodo2.tfg.com
addr=10.22.132.11 nodo3.tfg.com addr=10.22.132.12 nodo4.tfg.com
addr=10.22.122.13 balanceador.tfg.com addr=10.22.132.14 -u hacluster
```

Además, se puede observar que cada nodo usará como dirección IP para su comunicación en el clúster, la dirección IP que hace referencia a la red aislada de Control (10.22.132.0/24).

Una vez autenticado el usuario "*hacluster*", se procederá con la creación del propio clúster. Para ello, se hará uso de la orden "***pcs cluster setup***". En la creación del clúster se debe indicar: nombre del clúster, los nodos que lo van a conformar y las direcciones IP de cada nodo en el clúster. Además, como se ha hecho uso del flag "**--start**" se iniciarán los servicios que dan soporte al clúster en todos los nodos. Por lo tanto, para la creación de este clúster, se ejecutará la siguiente orden en el Nodo1 (Sólo hace falta ejecutar esta orden en un nodo del clúster):

```
$ pcs cluster setup cluster-tfg --start nodo1.tfg.com
addr=10.22.132.10 nodo2.tfg.com addr=10.22.132.11 nodo3.tfg.com
addr=10.22.132.12 nodo4.tfg.com addr=10.22.122.13
balanceador.tfg.com addr=10.22.132.14
```

Una vez ejecutada la orden anterior, se habrá creado el clúster llamado "**cluster-tfg**" que estará conformado por cinco nodos (Nodo1, Nodo2, Nodo3, Nodo4 y Balanceador-De-Carga).

Para que los servicios que dan soporte al clúster se inicien de manera automática en el arranque de cada nodo, sin necesidad de tener que iniciarlos

de manera manual, se ejecutará la siguiente orden desde el Nodo1 (Ver ***Ilustración 13***):

```
[root@nodo1 ~]# pcs cluster enable --all
nodo1.tfg.com: Cluster Enabled
nodo2.tfg.com: Cluster Enabled
nodo3.tfg.com: Cluster Enabled
nodo4.tfg.com: Cluster Enabled
balanceador.tfg.com: Cluster Enabled
```

Ilustración 13: Habilitación de los servicios que dan soporte al clúster

Por último, es muy recomendable que antes de realizar configuraciones en el clúster, se verifique su estado y se compruebe que está funcionando correctamente. Para ello se hará uso de la orden "***pcs cluster status"*** desde cualquiera de los nodos del clúster (Ver ***Ilustración 14***):

```
[root@nodo1 ~]# pcs cluster status
Cluster Status:
 Cluster Summary:
   * Stack: corosync
   * Current DC: nodo2.tfg.com (version 2.1.0-8.0.1.el8-7c3f660707) - partition with quorum
   * Last updated: Tue Jun 14 12:47:49 2022
   * Last change:  Tue Jun 14 12:47:07 2022 by hacluster via crmd on nodo2.tfg.com
   * 5 nodes configured
   * 0 resource instances configured
 Node List:
   * Online: [ balanceador.tfg.com nodo1.tfg.com nodo2.tfg.com nodo3.tfg.com nodo4.tfg.com ]

PCSD Status:
  nodo1.tfg.com: Online
  nodo2.tfg.com: Online
  nodo3.tfg.com: Online
  nodo4.tfg.com: Online
  balanceador.tfg.com: Online
```

Ilustración 14: Estado del clúster

16.11. Configuración de un mecanismo de aislamiento de nodos

En un clúster, a medida que aumenta la cantidad de nodos, aumenta también su disponibilidad, pero a su vez aumenta la posibilidad de que algún nodo tenga alguna falla en algún momento determinado y perjudique a los servicios ofrecidos por el clúster. Si la comunicación con un solo nodo del clúster fallara, los demás nodos del clúster deben poder restringir o liberar el acceso a

los recursos a los que el nodo fallido puede tener acceso. Para restringir este acceso, y poder aislar el nodo con fallas del clúster, se debe proporcionar un mecanismo externo, denominado **Fencing** o **Stonith**. Este método, aislará a los nodos con fallas del clúster, evitando que los mismos tengan acceso a los recursos del clúster. Para ello, se configurará un mecanismo de aislamiento de nodos basado en el host anfitrión. Para ver los detalles de los pasos para la creación de este mecanismo de aislamiento de nodos se puede consultar el Anexo XXV. Además, los pasos para la creación de este mecanismo de *fencing* y la información acerca de este método comentada anteriormente, se encuentra accesible en [47],[48].

16.12. Configuración de HAProxy como un recurso del clúster

Con el clúster creado y funcionando correctamente, se procederá a agregar el servicio HAProxy como un recurso del mismo. Recordamos, que el servicio HAProxy, es el encargado de ofrecer balanceo de carga y alta disponibilidad al servicio servicio Apache, servidor web en el que se basará el sistema de gestión de contenidos *Wordpress* que se instalará más adelante. Además, en apartados posteriores, se instalará la base de datos MAriaDB en los Nodos 3 y 4 y se configurará HAProxy para ofrecer equilibrio de carga en la misma, por lo cual, el servicio HAProxy es un servicio muy importante en la infraestructura construida. Por todo ello, se agregará el servicio HAProxy como un recurso del clúster, para que este servicio sea gobernado por el mismo. Para ello se ejecutará en cualquier nodo:

```
$ pcs resource create HAProxy-Service service:haproxy
```

Tras crear el recurso denominado ***HAProxy-Service***, se podrá desactivar (***systemctl disable haproxy***) y parar (***systemctl stop haproxy***) el servicio HAProxy que se ejecuta desde el nodo balanceador.tfg.com, ya que este servicio estará, ahora, gobernado por el clúster.

Una vez realizadas las configuraciones anteriores, se tendrá un problema, y es que el servicio HAProxy solamente está instalado y configurado en el nodo **balanceador.tfg.com**. Esto quiere decir, que si el nodo **Balanceador-De-Carga** no es el que inicia el recurso HAProxy, los demás nodos no podrán hacerlo, debido a que no está instalado ni configurado en ellos y fallará el arranque del mismo. Por lo tanto, se usará la orden "***pcs constraint***" para impedir que los nodos: nodo1.tfg.com, nodo2.tfg.com, nodo3.tfg.com y nodo4.tfg.com, puedan iniciar el recurso correspondiente al servicio HAProxy, ya que no está instalado en ellos [49].

```
$ pcs constraint location HAProxy-Service avoids nodo1.tfg.com
nodo2.tfg.com nodo3.tfg.com nodo4.tfg.com
```

Por último, es muy importante comprobar el estado del clúster (***pcs status***) y verificar que los recursos están iniciados correctamente y están siendo ofrecidos por alguno de los nodos del clúster (En este caso: xvmfence por el Nodo1 y HAProxy-Service por el Balanceador-De-Carga) (Ver ***Ilustración 15***):

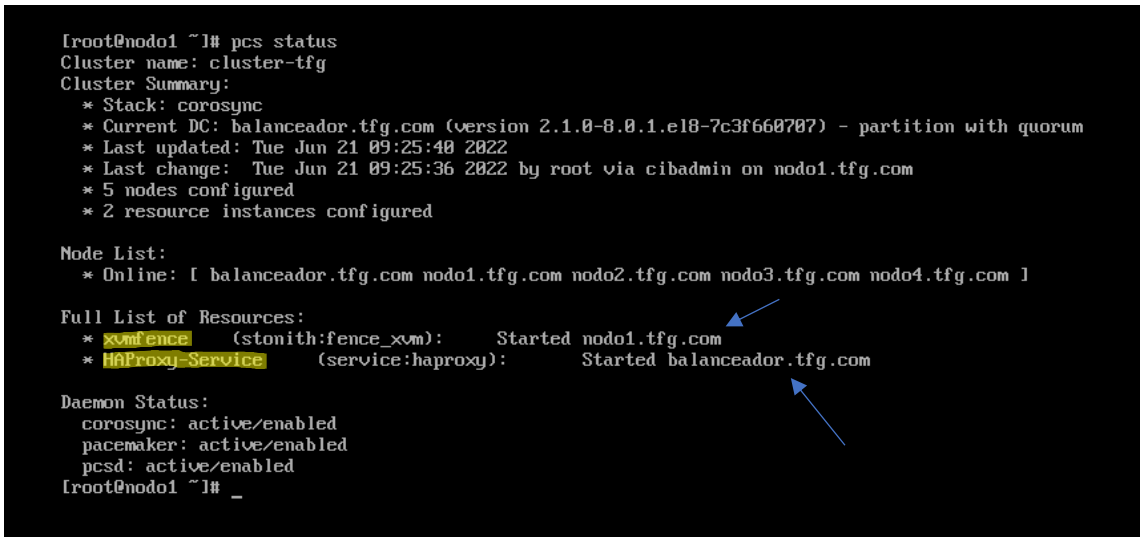

Ilustración 15: Estado del clúster y los recursos del mismo iniciados correctamente

16.13. Instalación y configuración de Galera Cluster

Una vez configurado el clúster, con sus respectivos recursos ejecutándose de manera correcta, se procederá a instalar, crear y configurar un clúster de bases de datos *MariaDB* en los nodos: **nodo3.tfg.com** y **nodo4.tfg.com**, con el fin de ofrecer replicación síncrona y alta disponibilidad, en la base de datos. Cabe recordar, que el fin de este proyecto es ofrecer un servicio web mediante *Wordpress* en alta disponibilidad y con balanceo de carga, y uno de los requisitos de este sistema de gestión de contenidos es el uso de una base de datos. Es por ello, que se hará uso de Galera Cluster para ofrecer una base de datos común, mediante la replicación síncrona de sus datos, en los Nodos 3 y 4 y que proporcione alta disponibilidad en el caso de que uno de los dos nodos (Nodos 3 y 4) no esté operativo. Para ello, en primer lugar, se deberá instalar todo el software necesario para **MariaDB Galera Cluster**, en los nodos **nodo3.tfg.com** y **nodo4.tfg.com** [50]:

```
$ dnf -y module install mariadb:10.3/galera
```

Como resultado de la ejecución de la orden anterior, se instalan los siguientes paquetes: **mariadb-server-galera**, **mariadb-server** y **galera**.

Una vez instalado “MariaDB Galera Cluster”, se procederá con la configuración del clúster de bases de datos tal y como se indica en [51]. Para ello, se deberán abrir una serie de puertos en el firewall de los nodos 3 y 4, tales como: 3306 (para las conexiones desde el cliente mariadb), 4567 (para la replicación entre los nodos en TCP y UDP), 4568 (para recibir los estados faltantes de los nodos) y 4444 (para recibir el estado y los datos de los nodos):

```
$ firewall-cmd --permanent --add-port=3306/tcp
```

```
$ firewall-cmd --permanent --add-port=4567/tcp
```

```
$ firewall-cmd --permanent --add-port=4568/tcp
```

```
$ firewall-cmd --permanent --add-port=4444/tcp
```

```
$ firewall-cmd --permanent --add-port=456/udp
```

```
$ firewall-cmd --reload
```

A continuación, se debe detener el servicio *MariaDB* tanto en el Nodo3 como en el Nodo4:

```
$ systemctl stop mariadb
```

Una vez se tenga el servicio MariaDB parado, se procederá a modificar el fichero de configuración de "Galera Cluster" en el nodo3 y en el Nodo4. Este fichero de configuración recibe el nombre de "***galera.cnf***" y está bajo el directorio ***"/etc/my.cnf.d".*** En este archivo, se deben localizar y modificar cuatro parámetros:

- **wsrep_cluster_name**: Se debe indicar el nombre del clúster de bases de datos MariaDB y debe ser el mismo para cada nodo.
- **wsrep_cluster_address**: Se debe indicar las direcciones IP de los dos nodos que conformarán el clúster de Galera. Además, este parámetro debe comenzar con la cadena "***gcomm://***".
- **wsrep_node_name**: Se debe indicar el nombre que recibirá el nodo del clúster de Galera que se esté configurando en ese momento. Este nombre puede ser cualquiera, pero es recomendable usar un nombre que identifique al nodo de manera adecuada.

- **wsrep_node_address**: Se debe indicar la dirección IP del nodo del clúster de Galera que se esté configurando en ese momento.

Por lo tanto, el fichero de configuración de Galera (***/etc/my.cnf.d/galera.cnf***) en el Nodo3, tendrá la siguiente información en los parámetros comentados anteriormente (Ver ***Ilustración 16***):

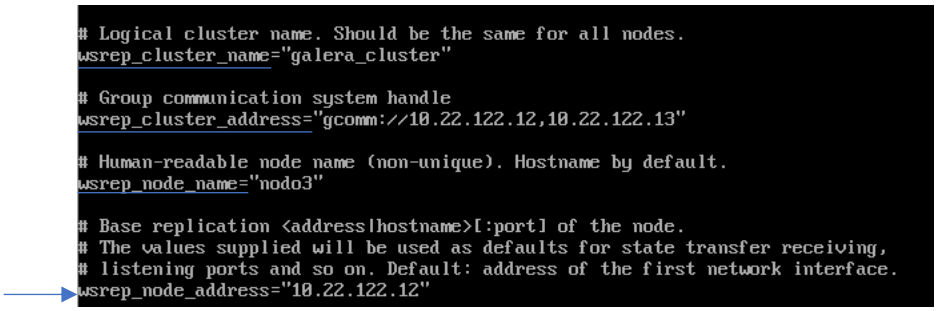

Ilustración 16: Fichero de configuración de Galera en el Nodo3

Por otro lado, el fichero de configuración de Galera (***/etc/my.cnf.d/galera.cnf***) en el Nodo4, tendrá la siguiente información en los parámetros comentados anteriormente (Ver ***Ilustración 17***):

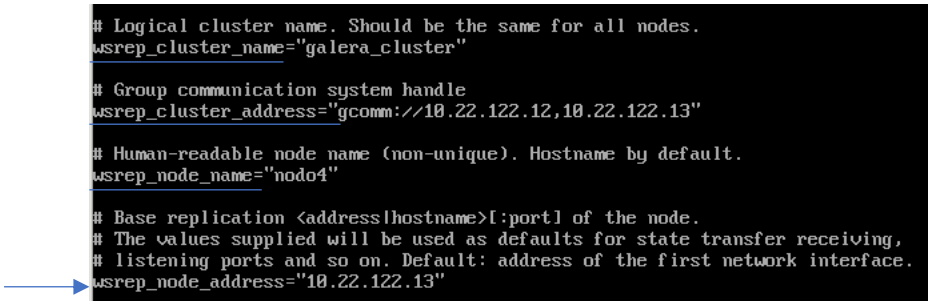

Ilustración 17: Fichero de configuración de Galera en el Nodo4

Nota: El fichero de configuración ***"/etc/my.cnf.d/galera.cnf***" contiene más parámetros de configuración, pero en las ilustraciones 16 y 17, tan sólo se han mostrado los parámetros que se han modificado, ya que los demás parámetros se han dejado por defecto.

Una vez llegados a este punto, se tendrá configurado de manera satisfactoria los dos nodos que conforman el clúster de bases de datos MariaDB, proporcionado por los nodos: **nodo3.tfg.com** y **nodo4.tfg.com**. Lo siguiente en realizar será, con el servicio MariaDB parado en ambos nodos (3 y 4), se debe inicializar el clúster de bases de datos MariaDB de Galera, ejecutando el siguiente comando únicamente desde el Nodo3 (Es muy importante tener en cuenta que este comando tan solo se puede ejecutar en un nodo):

```
$ galera_new_cluster
```

Si la salida del comando anterior no muestra nada por pantalla, quiere decir que se ha ejecutado con éxito y que el Nodo3 se ha unido satisfactoriamente al clúster, y que además el servicio MariaDB ha sido iniciado automáticamente por Galera en el Nodo3 (Ver status de *mariadb* en el Nodo3).

Además, se puede comprobar el tamaño del clúster de Galera recién creado, consultando la variable "**wsrep_cluster_size**" de la siguiente manera desde el Nodo3 (Ver ***lustración 18***):

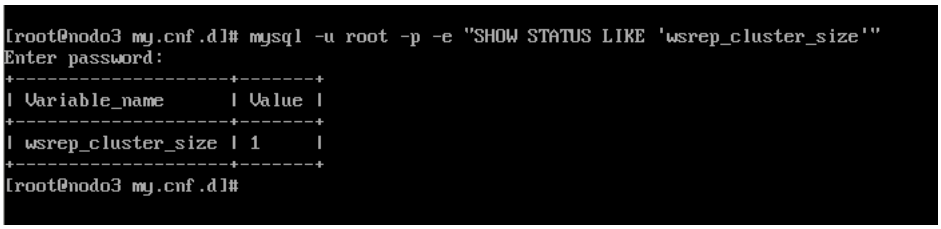

Ilustración 18: Tamaño del clúster de Galera con un nodo

Como se puede observar en la figura anterior, se ha consultado la variable que indica el tamaño del clúster de Galera, estableciendo una conexión mediante el cliente MariaDB y usando el usuario root (contraseña en blanco por defecto). Como se observa en la imagen, el resultado de la consulta ha sido el de "1", debido a que tan sólo un nodo se ha unido al clúster hasta el momento (Nodo3).

Acto seguido, se procederá a unir al clúster de Galera recién creado, al Nodo4. Para que el Nodo4 se una al clúster, tan solo es necesario iniciar el servicio MariaDB desde el Nodo4:

```
$ systemctl start mariadb
```

Nuevamente, se debe consultar desde alguno de los dos nodos, la variable "**wsrep_cluster_size**", para verificar que el Nodo4 se ha unido satisfactoriamente al clúster de Galera (Ver ***Ilustración 19***):

```
[root@nodo4 my.cnf.d]# mysql -u root -p -e "SHOW STATUS LIKE 'wsrep_cluster_size'"
Enter password:
+--------------------+-------+
| Variable_name      | Value |
+--------------------+-------+
| wsrep_cluster_size | 2     |
+--------------------+-------+
```

Ilustración 19: Tamaño del clúster de Galera con dos nodos

Como se observa en la imagen anterior, ahora el tamaño del clúster es de "2" ya que se ha unido el Nodo4 a él.

Nota: Es muy importante mencionar, que una vez que se apaguen o reinicien los nodos 3 y 4, el clúster de Galera desaparecerá y habrá que volver a inicializarlo cuando se arranquen las máquinas, de forma manual. Para volver a inicializar el clúster de Galera, se realizará de la misma manera que se realizó anteriormente, pero hay que saber desde que nodo se debe inicializar el clúster mediante el comando "*galera_new_cluster*", ya que no siempre será el Nodo3 el que inicialice el clúster de Galera. Para saber desde que nodo se debe inicializar el clúster, se debe observar el contenido del fichero ***"/var/lib/mysql/grastate.dat***" y localizar la variable "***safe_to_bootstrap***". Si esta variable está a "1" en alguno de los nodos, entonces ese será el nodo que debe inicializar el clúster mediante "*galera_new_cluster*". Si la variable está a "0" en alguno de los nodos, entonces quiere decir que ese no es el nodo que debe inicializar el clúster de Galera.

Además, una vez inicializado el clúster nuevamente, se deberá unir al nodo restante (nodo 3 o 4) al clúster de Galera, tan solo iniciando el servicio "*mariadb*".

Por último, una vez que se tenga el clúster de Galera totalmente configurado, con sus dos nodos unidos satisfactoriamente a él, se debe verificar la replicación de datos entre los dos nodos de clúster de Galera (Nodos 3 y 4) y la alta disponibilidad de la base de datos ofrecida por ellos. Para ver las pruebas de la replicación síncrona de los datos de la base de datos y su alta disponibilidad, se puede consultar el Anexo XXVI.

16.14. Configuración de HAProxy para proporcionar balanceo de carga a los nodos de Galera Cluster

Una vez que se disponga de un clúster de bases de datos MariaDB de Galera, que proporciona replicación síncrona y alta disponibilidad entre los nodos que conforman el clúster (Nodos 3 y 4), se procederá a configurar HAProxy para proporcionar, además, balanceo de carga en las peticiones que realicen los potenciales clientes a la base de datos MariaDB del Nodo3 y Nodo4. Para ello, se seguirán los pasos descritos en [52], y en primer lugar se deberá activar un booleano de SELinux en el nodo **balanceador.tfg.com**, para permitir que HAProxy pueda conectarse a todos los puertos TCP:

```
$ setsebool -P haproxy_connect_any on
```

Acto seguido, se deberá modificar el fichero ***"/etc/haproxy/haproxy.cfg***" del nodo **balanceador.tfg.com**, para indicarle a HAProxy que proporcione balanceo de carga a los nodos que conforman el clúster de Galera. Para ello, se hará uso del algoritmo "**roundrobin**" para distribuir las peticiones a la base de datos MariaDB, se habilitará la función ***keepalive*** para mantener las conexiones TCP (tcpka), se escucharán las peticiones entrantes mediante la dirección IP

192.168.122.209 y el puerto **3306** y se definirán los nodos (con sus direcciones IP) que HAProxy debe usar en las conexiones de enrutamiento. Para todo ello, se agregará al final del fichero "**haproxy.cfg**", la siguiente sección (Ver ***Ilustración 20***).

```
# Node that offer Galera Cluster with load Balancing
listen galera
    bind 192.168.122.209:3306
    balance roundrobin
    mode tcp
    option tcpka
    server nodo3 10.22.122.12 check port 3306 weight 1
    server nodo4 10.22.122.13 check port 3306 weight 1
```

Ilustración 20: Configuración del fichero HAProxy para proporcionar balanceo de carga a la BD de los Nodos 3 y 4

Por último, se deberá agregar el puerto 3306 (correspondiente al servicio mysql), al cortafuegos del nodo **Balanceador-De-carga**:

```
$ firewall-cmd --permanent --add-port=3306/tcp
```

```
$ firewall-cmd --reload
```

Una vez realizado lo anterior, se dispondrá de balanceo de carga en las peticiones que se realicen a la base de datos ofrecida en alta disponibilidad por los nodos: **nodo3.tfg.com** y **nodo4.tfg.com**. Para comprobar el equilibrio de carga en las peticiones a la base de datos, se podrá consultar el Anexo XXVII.

16.15. Configuración de la interfaz gráfica de estadísticas de HAProxy

HAProxy permite visualizar el comportamiento general de las secciones *frontend* y *backend* por medio de una interfaz web gráfica. Con esta interfaz gráfica, se podrán consultar algunas estadísticas como: el tiempo que

tardó una petición/respuesta, el encolamiento, las sesiones iniciadas, los nodos operativos e inoperativos, etc. Para la configuración de esta interfaz gráfica para monitorear estadísticas del servicio HAProxy [53] proporcionado, se deberá agregar al final del fichero ***"/etc/haproxy/haproxy.cfg"*** del nodo balanceador.tfg.com, la siguiente sección:

```
# Stats for HAProxy
listen stats
    bind 192.168.122.209:9000
    mode http
    stats enable
    stats uri /haproxy_stats
    stats realm Haproxy\ Statistics
    stats auth haproxy:haproxy
    stats admin if TRUE
```

Ilustración 21: Configuración del fichero de HAProxy para agregar la interfaz gráfica de estadísticas

Observando el contenido de la sección de estadísticas del fichero "**haproxy.cfg**" de la imagen anterior, se puede observar que el acceso a la interfaz gráfica de estadísticas de HAProxy se hará a través de la dirección IP cliente **192.168.122.209** y el puerto **9000**. Además, se hará por medio del protocolo **http**, el usuario con permisos de administrador **haproxy**, la contraseña **haproxy** y la URL **"/haproxy_stats**".

Antes de poder acceder a esta interfaz gráfica de estadísticas, se debe abrir el puerto **9000** en el cortafuegos del nodo **balanceador.tfg.com**. Este puerto es por el cual se va a acceder a la interfaz gráfica de estadísticas, por lo que se debe abrir en el firewall:

```
$ firewall-cmd --permanent --add-port=9000/tcp
```

```
$ firewall-cmd --reload
```

Por último, se pueden consultar algunos detalles del funcionamiento de esta interfaz gráfica de estadísticas en el Anexo XXVIII.

16.16. Instalación y configuración del componente PHP del software LAMP

En apartados anteriores, se ha creado el clúster, se ha instalado y configurado el software necesario en cada nodo y se ha proporcionado alta disponibilidad y balanceo de carga tanto al servicio web Apache como a la base de datos MariaDB. Tras esto, se procederá a instalar el componente que falta del software LAMP (Linux, Apache, MySQL y PHP), software necesario para poder instalar y configurar el sistema de gestión de contenidos *Wordpress* a ofrecer en alta disponibilidad y con balanceo de carga. Este software LAMP es indispensable, ya que, sin él, *Wordpress* no funcionará correctamente. En secciones anteriores, ya se ha efectuado la instalación de Apache en los Nodos 1 y 2, y además se ha instalado la base de datos MariaDB en los Nodos 3 y 4. Por lo que sabiendo que se está trabajando sobre una distribución de Linux (otro componente de la pila LAMP), solo es necesario instalar PHP, para completar la instalación y configuración del servidor LAMP.

Para la instalación y configuración de PHP, se instalará tanto en el Nodo1 como en el Nodo2, la versión que más interesa del mismo (7.4 en este aso) y sus extensiones necesarias [54]:

```
$ dnf module install php:7.4
```

```
$ dnf -y install php php-fpm php-mysqlnd php-opcache php-gd php-xml php-mbstring
```

Una vez instalado **PHP 7.4** y sus extensiones en los nodos: **nodo1.tfg.com** y **nodo2.tfg.com**, se debe habilitar e iniciar "***php-fpm***" en los Nodos 1 y 2 y verificar que se encuentra activo sin errores (status):

```
$ systemctl enable php-fpm
```

```
$ systemctl start php-fpm
```

```
$ systemctl status php-fpm
```

Una vez habilitado e iniciado correctamente "***php-fpm***", es muy recomendable reiniciar el servidor web *Apache* en los nodos 1 y 2 (***systemctl restart httpd***) y verificar con el comando "**php -v**" que se ha instalado la versión correcta de PHP en los Nodos 1 y 2 (Ver ***Ilustración 22***):

Ilustración 22: Versión de PHP instalada en los Nodos 1 y 2

Nota: Es muy importante verificar los servicios y puertos abiertos en el firewall de los Nodos 1 y 2 ("***firewall-cmd --list-all***") y comprobar que entre estos se encuentran los servicios "http" y "https".

Por último, se realizarán una serie de configuraciones en los Nodos 1 y 2 relativas a PHP para su correcto funcionamiento. Se podrá observar estas configuraciones consultando el Anexo XXIX.

16.17. Instalación y configuración de *Wordpress*

Una vez configurado el software LAMP, se procederá a instalar el sistema de gestión de contenidos que será ofrecido en alta disponibilidad y con balanceo de carga, *Wordpress [55]*. Para ello, en primer lugar, se descargará la última versión de *Wordpress* desde su página oficial mediante la utilidad "*curl*" en los Nodos 1 y 2:

```
$ curl -O https://wordpress.org/latest.tar.gz
```

Acto seguido, se descomprimirá el paquete de la última versión de *Wordpress* instalado con el comando anterior, en el directorio ***"/var/www/html***" de los Nodos 1 y 2:

```
$ tar zxf latest.tar.gz -C /var/www/html --strip 1
```

Una vez descomprimido el paquete con todas las carpetas necesarias de *Wordpress*, se cambiarán los permisos de usuario y de grupo a la carpeta que aloja los archivos de Wordpress (***/var/www/html***) y a todos sus descendientes ***(/var/www/html/*).*** Para ello, se debe ejecutar en los Nodos 1 y 2***:***

```
$ chown apache. -R /var/www/html
```

Acto seguido, se creará el directorio de descargas de Wordpress "**uploads**", tanto en el Nodo 1 como en el Nodo2 y se ajustarán sus permisos:

```
$ mkdir /var/www/html/wp-content/uploads
```

```
$ chown apache:apache /var/www/html/wp-content/uploads
```

Tras cambiar los permisos de los archivos de *Wordpress*, es recomendable verificarlos mediante el comando "***ls -l***" y establecer los contextos de SELinux apropiados a la carpeta ***"/var/www/html***" y sus descendientes:

```
$ chcon -t httpd_sys_rw_content_t /var/www/html -R
```

Además, para permitir que Apache se pueda conectar a bases de datos externas, se debe activar el siguiente booleano de SELinux en los nodos: **nodo1.tfg.com** y **nodo2.tfg.com**:

```
$ setsebool -P httpd_can_network_connect_db 1
```

Como la configuración por defecto del servidor *Apache* ignora los archivos "***.htacces"*** y la configuración del módulo **Rewrite** se carga a través de este archivo, es necesario configurar *Apache* para permitir estos archivos. Para ello, se creará un archivo independiente denominado "***wordpress.conf***" en el directorio ***"/etc/httpd/conf.d/"*** de los nodos 1 y 2 con el siguiente contenido (Ver ***Ilustración 23***)

```
[root@nodo1 ~]# cat /etc/httpd/conf.d/wordpress.conf
<Directory /var/www/html>
        AllowOverride All
</Directory>
[root@nodo1 ~]# _
```

Ilustración 23: Contenido del fichero wordpress.conf de los Nodos 1 y 2

Tras esto, se debe añadir un host virtual para el sitio web de *Wordpress* al final de los ficheros de configuración de Apache ***(/etc/httpd/conf/httpd.conf***) de los nodos 1 y 2. Tras añadir el host virtual, el fichero de configuración de Apache en los Nodos 1 y 2, lucirá de la siguiente manera (Ver ***Ilustración 24***):

```
<VirtualHost *:80>
        ServerAdmin wordpress@wordpress.com
        DocumentRoot /var/www/html
        ServerName wordpress.com
        ServerAlias www.wordpress.com
        ErrorLog /var/log/httpd/wordpress-error-log
        CustomLog /var/log/httpd/wordpress-access-log common
</VirtualHost>
```

Ilustración 24: VirtualHost en el fichero de configuración de Apache de los nodos 1 y 2

Por último, se debe reiniciar el servicio Apache (***systemctl restart httpd***) en los nodos: nodo1.tfg.com y nodo2.tfg.com.

16.18. Creación de una base de datos y un usuario para *Wordpress*

Siguiendo la misma documentación de *Oracle Linux* que en el apartado anterior, se procederá a crear una base de datos llamada "*wordpress*" y un usuario llamado también "*wordpress*" con privilegios sobre esta base de datos. Esta base de datos y este usuario serán usados para el sitio web de *Wordpress* a ofrecer en alta disponibilidad y con balanceo de carga. Para la creación de la base de datos y del usuario para Wordpress, se ha usado el cliente de MariaDB desde el Nodo3 como se muestra en la siguiente ilustración:

```
[root@nodo3 mysql]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 10
Server version: 10.3.32-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database wordpress;
Query OK, 1 row affected (0.011 sec)

MariaDB [(none)]> create user wordpress@'%' identified by 'wordpress';
Query OK, 0 rows affected (0.017 sec)

MariaDB [(none)]> grant all privileges on wordpress.* to wordpress@'%' identified by 'wordpress';
Query OK, 0 rows affected (0.011 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.010 sec)

MariaDB [(none)]> exit
Bye
```

Ilustración 25: Creación de la BD y del usuario para Wordpress

Una vez creada la base de datos de *Wordpress* desde el Nodo3, la misma también se encontrará replicada en el Nodo4 (Ver ***Ilustración 26***), gracias al clúster de Galera conformado por los Nodos 3 y 4.

```
[root@nodo4 mysql]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 11
Server version: 10.3.32-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| prueba_replicacion |
| wordpress          |
+--------------------+
```

Ilustración 26: Base de datos Wordpress replicada en el Nodo4

16.19. Instalación de *Wordpress* desde el navegador

Una vez llegados a este punto, disponemos de todo el software LAMP que necesita *Wordpress* configurado de manera adecuada en los diferentes nodos. Para comenzar a usar Wordpress, solo quedará preparar este servicio web desde el navegador. Para ello, se seguirán una serie de pasos que se pueden consultar en detalle en el Anexo XXX.

Una vez que se tiene configurado *Wordpress* en el navegador, ya se tendrá acceso al sitio web de *Wordpress* con alta disponibilidad y con balanceo de carga, ya que este sistema de gestión de contenidos está basado en Apache y este servidor web está ofrecido en alta disponibilidad y con balanceo de carga mediante la dirección IP cliente **192.168.122.209**. Además, para verificar que este servicio se está ofreciendo en alta disponibilidad y con balanceo de carga, se borrará el fichero ***"/var/www/html/index.html"*** correspondiente a la página principal de Apache en los nodos 1 y 2 (**rm -rf /var/www/html/index.html**) y se procederá a modificar el fichero "***/var/www/html/index.php***" en los nodos 1 y 2, para saber en todo momento que nodo está ofreciendo el servicio web *Wordpress* en alta disponibilidad y con balanceo de carga. Para ello, se deberá agregar la siguiente línea en el fichero **index.php** del Nodo1:

```
echo "<h1 style='text-align:center;'>SOY EL NODO1</h1>";
```

Y agregar la siguiente línea en el fichero **index.php** del Nodo2:

```
echo "<h1 style='text-align:center;'>SOY EL NODO2</h1>";
```

Por último, una vez modificado el fichero **index.php** correspondiente a la página principal de *Wordpress* tanto en el Nodo1 como en el Nodo2, se podrá acceder mediante de la dirección IP cliente **192.168.122.209** y el navegador del host anfitrión, al servicio web de *Wordpress* ofrecido. Para ver los detalles del

sitio web de *Wordpress* y verificar el balanceo de carga y la alta disponibilidad en él, se puede consultar el Anexo XXXI.

16.20. Configuración del servidor y los clientes iSCSI para proporcionar un almacenamiento compartido

Llegados a este punto, se tiene una infraestructura de clúster conformada por cinco nodos, donde dos de ellos (nodos 1 y 2) ofrecen el servicio *Wordpress* basado en Apache, otros dos (nodos 3 y 4) ofrecen la base de datos MariaDB usada por Wordpress en alta disponibilidad y un nodo (Balanceador-De-Carga) es el encargado de ofrecer alta disponibilidad al servicio web *Wordpress* y equilibrio de carga al mismo y a su base de datos.

Con este escenario, se procederá a crear y configurar un almacenamiento compartido externo mediante la tecnología iSCSI, que ofrezca un espacio común de almacenamiento a los cinco nodos del clúster. Para ello, se hará uso del nodo **iscsi.tfg.com**, que actuará como servidor externo de iSCSI. Cabe recordar, que este nodo ("Almacenamiento-iSCSI"), es una máquina virtual de KVM que se ha creado en la sección 14.5 del capítulo 14, no forma parte del clúster, solamente dispone de dos interfaces de red (Red NAT y Red de Almacenamiento) y además tiene un segundo disco de almacenamiento de 10GB. Este disco es visible desde la máquina virtual **iscsi.tfg.com** como "**vdb**" y está sin particionar (Ver **Ilustración 27):**

```
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0          11:0    1 1024M  0 rom
vda         251:0    0   15G  0 disk
├─vda1      251:1    0    1G  0 part /boot
└─vda2      251:2    0   14G  0 part
  ├─ol-root 252:0    0 12,5G  0 lvm  /
  └─ol-swap 252:1    0  1,5G  0 lvm  [SWAP]
vdb         251:16   0   10G  0 disk
```

Ilustración 27: Discos de almacenamiento del servidor iSCSI

Acto seguido, se procederá con la configuración del almacenamiento compartido iSCSI tanto en el nodo del servidor iSCSI (nodo iscsi.tfg.com) como

en los nodos clientes (nodos del clúster). Para ello, se seguirá la documentación proporcionada por el sitio web *IT'zGeek* en [56].

En primer lugar, se debe instalar el servicio ***iSCSI*** en el nodo **iscsi.tfg.com**, ya que será el nodo encargado de ofrecer el almacenamiento compartido (nodo target) a los nodos del clúster. Para ello, se ejecutará en él:

```
$ dnf -y install targetcli
```

Acto seguido, se instalará el software correspondiente al cliente iSCSI en todos los nodos del clúster: nodo1.tfg.com, nodo2.tfg.com, nodo3.tfg.com, nodo4.tfg.com y balanceador.tfg.com:

```
$ dnf -y install iscsi-initiator-utils
```

Tras la ejecución del comando anterior, se habrá generado un identificador en cada uno de los nodos del clúster (nodos cliente), que permitirá identificar a cada uno de los nodos cliente en el nodo servidor (iscsi.tfg.com). Este identificador se encuentra en el fichero ***"/etc/iscsi/initiatorname.iscsi"*** de los nodos clientes (nodos initiator), y se puede modificar con el propósito de identificar lo mejor posible a cada nodo *initiator*. Por ello, el contenido del fichero ***"/etc/iscsi/initiatorname.iscsi"*** de cada nodo del clúster tiene el siguiente aspecto:

```
InitiatorName=iqn.1988-12.com.oracle:Nombre_del_nodo
```

Donde la cadena resaltada en color gris "**Nombre_del_nodo**" hace referencia al nombre descriptivo de cada nodo. Por ejemplo, en el Nodo1, la cadena "**Nombre_del_nodo**" sería nodo1, en el Nodo2 sería nodo2, en el Balanceador-De-Carga sería balanceador, etc.

Una vez identificados los nodos *initiator* e instalado el servidor *iSCSI* en el nodo **iscsi.tfg.com** y el cliente **iSCSI** en los nodos del clúster, se procederá a crear un espacio de almacenamiento compartido que use todo el espacio libre del segundo disco de la MV **iscsi.tfg.com** (10GB). Para ello, se creará en primera instancia un volumen físico sobre el disco virtual "**vdb**" del nodo **iscsi.tfg.com**:

```
$ pvcreate /dev/vdb
```

Luego, se creará un grupo de volúmenes llamado "vg_iscsi" que albergue al volumen físico creado anteriormente:

```
$ vgcreate vg_iscsi /dev/vdb
```

Y, por último, se creará un volumen lógico llamado "lv_almacenamiento_iscsi" que use el 100% del espacio libre del grupo de volúmenes creado anteriormente (10GB):

```
$ lvcreate -l 100%FREE -n lv_almacenamiento_iscsi vg_iscsi
```

Este volumen lógico será el espacio de almacenamiento compartido que será importado posteriormente a los nodos cliente (nodos del clúster).

En la **Ilustración 28**, se pueden ver los detalles de los volúmenes creados en el nodo Almacenamiento-iSCSI:

```
[root@iscsi ~]# pvscan
  PV /dev/vdb    VG vg_iscsi        lvm2 [<10,00 GiB / 0    free]
  PV /dev/vda2   VG ol              lvm2 [<14,00 GiB / 0    free]
  Total: 2 [23,99 GiB] / in use: 2 [23,99 GiB] / in no VG: 0 [0   ]
[root@iscsi ~]# vgscan
  Found volume group "vg_iscsi" using metadata type lvm2
  Found volume group "ol" using metadata type lvm2
[root@iscsi ~]# lvscan
  ACTIVE            '/dev/vg_iscsi/lv_almacenamiento_iscsi' [<10,00 GiB] inherit
  ACTIVE            '/dev/ol/swap' [1,50 GiB] inherit
  ACTIVE            '/dev/ol/root' [<12,50 GiB] inherit
```

Ilustración 28: Volúmenes físicos, grupos de volúmenes y volúmenes lógicos del nodo iscsi.tfg.com

A continuación, se usará la utilidad de línea de comandos "**targetcli**" desde el nodo **iscsi.tfg.com**, con el fin de crear un objeto de almacenamiento de bloque que haga uso del volumen lógico de 10GB creado anteriormente. A través de esta utilidad, también se creará el identificador (IQN) del nodo iscsi.tfg.com. Tras esto, se creará una lista de control de acceso (ACL) para cada nodo *initiator* (nodos del clúster) con el propósito de otorgar acceso exclusivo a un objetivo específico y se creará la unidad LUN que será exportada a los nodos cliente (Ver ***Ilustración 29***)

```
[root@iscsi ~]# targetcli
targetcli shell version 2.1.53
Copyright 2011-2013 by Datera, Inc and others.
For help on commands, type 'help'.

/> cd /backstores/block
/backstores/block> create iscsi_almacenamiento_compartido /dev/vg_iscsi/lv_almacenamiento_iscsi
Created block storage object iscsi_almacenamiento_compartido using /dev/vg_iscsi/lv_almacenamiento_iscsi.
/backstores/block> cd /iscsi
/iscsi> create
Created target iqn.2003-01.org.linux-iscsi.iscsi.x8664:sn.bef0d94d3001.
Created TPG 1.
Global pref auto_add_default_portal=true
Created default portal listening on all IPs (0.0.0.0), port 3260.
/iscsi> cd iqn.2003-01.org.linux-iscsi.iscsi.x8664:sn.bef0d94d3001/tpg1/acls
/iscsi/iqn.20...001/tpg1/acls> create iqn.1988-12.com.oracle:nodo1
Created Node ACL for iqn.1988-12.com.oracle:nodo1
/iscsi/iqn.20...001/tpg1/acls> create iqn.1988-12.com.oracle:nodo2
Created Node ACL for iqn.1988-12.com.oracle:nodo2
/iscsi/iqn.20...001/tpg1/acls> create iqn.1988-12.com.oracle:nodo3
Created Node ACL for iqn.1988-12.com.oracle:nodo3
/iscsi/iqn.20...001/tpg1/acls> create iqn.1988-12.com.oracle:nodo4
Created Node ACL for iqn.1988-12.com.oracle:nodo4
/iscsi/iqn.20...001/tpg1/acls> create iqn.1988-12.com.oracle:balanceador
Created Node ACL for iqn.1988-12.com.oracle:balanceador
/iscsi/iqn.20...001/tpg1/acls> cd /iscsi/iqn.2003-01.org.linux-iscsi.iscsi.x8664:sn.bef0d94d3001/tpg1/luns
/iscsi/iqn.20...001/tpg1/luns> create /backstores/block/iscsi_almacenamiento_compartido
Created LUN 0.
Created LUN 0->0 mapping in node ACL iqn.1988-12.com.oracle:balanceador
Created LUN 0->0 mapping in node ACL iqn.1988-12.com.oracle:nodo4
Created LUN 0->0 mapping in node ACL iqn.1988-12.com.oracle:nodo3
Created LUN 0->0 mapping in node ACL iqn.1988-12.com.oracle:nodo2
Created LUN 0->0 mapping in node ACL iqn.1988-12.com.oracle:nodo1
/iscsi/iqn.20...001/tpg1/luns> _
```

Ilustración 29: Creación mediante targetcli de los dispositivos iSCSI

Una vez realizado todos los pasos anteriores, se puede ver todos los dispositivos iSCSI desde la consola interactiva del comando "**targetcli**" haciendo uso de la orden "**ls**" desde la raíz ("/"). Además, es muy importante guardar todos los cambios realizados con la orden "**saveconfig**" desde la consola interactiva de la utilidad "**targetcli**" (Ver ***Ilustración 30***):

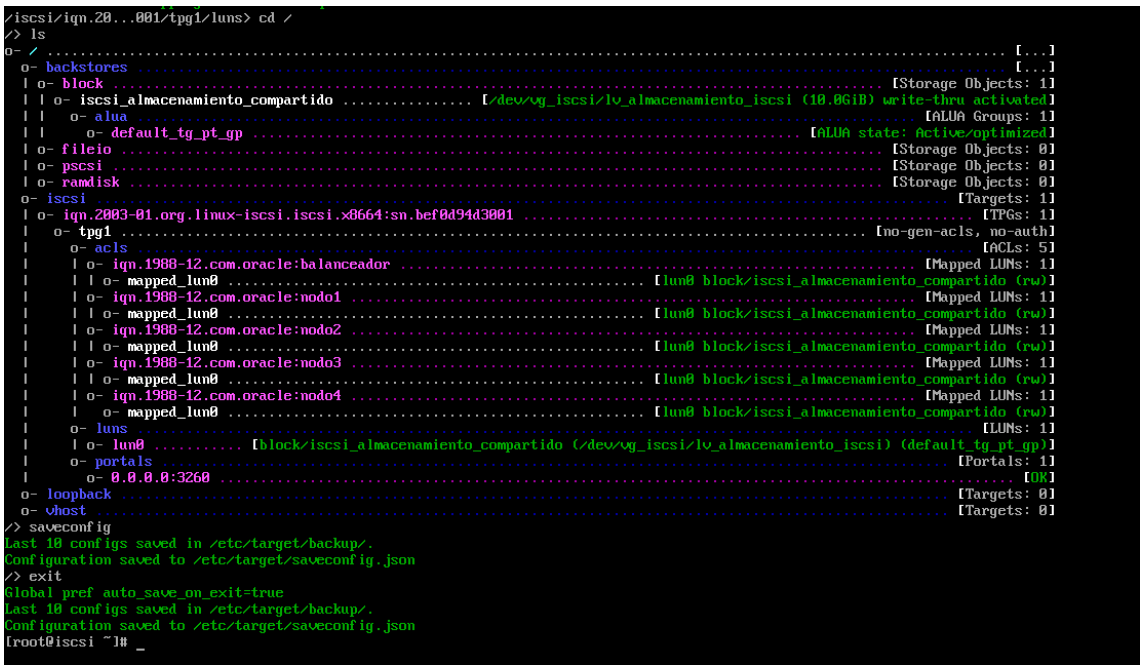

Ilustración 30: Listado de los dispositivos iSCSI creados

Para finalizar con la configuración del servidor iSCSI, se debe habilitar e iniciar el servicio "**target**" en el nodo **iscsi.tfg.com**:

```
$ systemctl enable target
```

```
$ systemctl start target
```

Y se debe configurar el cortafuegos del nodo Almacenamiento-iSCSI para permitir el tráfico iSCSI mediante el puerto 3260 (puerto por el cual escucha el portal de red):

```
$ firewall-cmd --permanent --add-port=3260/tcp
```

```
$ firewall-cmd --reload
```

Una vez configurado el servidor iSCSI para que pueda exportar un disco compartido a los diferentes nodos del clúster, se debe descubrir este

almacenamiento compartido exportado por el nodo **iscsi.tfg.com** (LUN), en todos los nodos del clúster. Para ello, en todos los nodos del clúster y haciendo uso la dirección IP de almacenamiento del nodo iscsi.tfg.com, se descubre le objetivo con la siguiente orden:

```
$ iscsiadm -m discovery -t st -p 10.22.122.15
```

El comando anterior genera como salida el IQN del nodo **iscsi.tfg.com**, el cuál es "**iqn.2003-01.org.linux-iscsi.iscsi.x8664:sn.bef0d94d3001**"

Ahora, se debe iniciar sesión en el almacenamiento compartido en todos los nodos del clúster (Nodos 1, 2, 3, 4 y Balanceador-De-Carga), conectándonos al IQN proporcionado en la ejecución del comando anterior:

```
$ iscsiadm -m mode -T iqn.2003-01.org.linux-
iscsi.iscsi.x8664:sn.bef0d94d3001 -p 10.22.122.15 -l
```

Una vez que se haya descubierto el almacenamiento compartido y se haya iniciado sesión en él, se debe iniciar y habilitar en el arranque de cada nodo del clúster, el servicio "**iscsid**":

```
$ systemctl enable --now iscsid
```

Por último, se puede verificar que, en todos los nodos del clúster, el almacenamiento compartido de 10GB proporcionado por el servidor iSCSI es visible como un disco llamado "**sda**". Para ello se puede hacer uso de la utilidad "**lsblk**" (Ver ***Ilustración 31***):

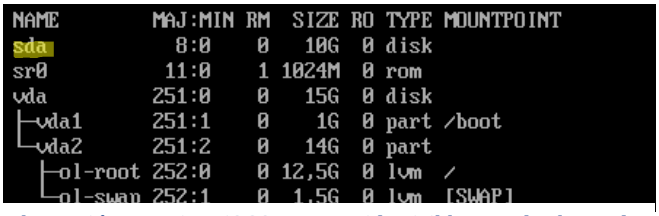

Ilustración 31: Disco iSCSI compartido visible en todos los nodos del clúster

16.21. Configuración del clúster OCFS2 para ofrecer almacenamiento compartido y síncrono

Con todas las configuraciones anteriores, los cinco nodos del clúster disponen de un disco común como almacenamiento compartido iSCSI, pero no se puede garantizar los accesos concurrentes de los diferentes nodos al almacenamiento compartido. Es por ello, que en este apartado se creará un clúster haciendo uso del sistema de archivos distribuido OCFS2 nativo de Oracle, para controlar el acceso concurrente de los diferentes nodos del clúster al almacenamiento compartido proporcionado por el servidor iSCSI externo.

Antes de comenzar a configurar el clúster OCFS2, en [57] se recomienda deshabilitar SELinux para trabajar con OCFS2, incluso se considera como una buena práctica. Por ello, se procederá a establecer la variable "**SELINUX**" del fichero "***/etc/selinux/config***" de los nodos **nodo1.tfg.com** y **nodo3.tfg.com** a "**permissive**".

Una vez con SELinux en modo permisivo en los nodos 1 y 3, se procederá con la configuración clúster OCFS2 siguiendo la documentación oficial de Oracle [58].

En primer lugar, se deberá configurar el cortafuegos de todos los nodos del clúster (Nodos 1, 2, 3 ,4 y Balanceador) para permitir el acceso a la interfaz que usará el clúster OCFS2 para la comunicación privada. De manera predeterminada, el clúster OCFS2 usa TCP como UDP a través del puerto 7777, por lo que se debe abrir este puerto en el Firewall de los nodos del clúster:

```
$ firewall-cmd --permanent --add-port=7777/tcp
```

```
$ firewall-cmd --permanent --add-port=7777/udp
```

```
$ firewall-cmd --reload
```

Acto seguido, se instalarán las utilidades para la creación del clúster OCFS2 y una versión de UEK, en los nodos 1, 2, 3, 4 y Balanceo, para disponer de versiones software compatibles de estos dos paquetes. Para ello, se ejecutará en cada nodo del clúster:

```
$ dnf -y install kernel-uek ocfs2-tools
```

Una vez instalado todo el software necesario en cada nodo, se continuará con la creación del clúster OCFS2. La creación del clúster OCFS2 se puede llevar a cabo desde solamente uno de los nodos que lo conforman. Por lo tanto, se ejecutará desde el Nodo1 la siguiente orden que crea el clúster OCFS2 llamado "**clusterOCFS2**":

```
$ o2cb add-cluster clusterOCFS2
```

Tras la creación del clúster OCFS2, se deben añadir los diferentes nodos que lo conformarán y las direcciones IP que usarán. Para ello, se añadirán los nodos al clúster ejecutando desde el Nodo1:

```
$ o2cb add-node clusterOCFS2 nodo1.tfg.com --ip 10.22.122.10
```

```
$ o2cb add-node clusterOCFS2 nodo2.tfg.com --ip 10.22.122.11
```

```
$ o2cb add-node clusterOCFS2 nodo3.tfg.com --ip 10.22.122.12
```

```
$ o2cb add-node clusterOCFS2 nodo4.tfg.com --ip 10.22.122.13
```

```
$ o2cb add-node clusterOCFS2 balanceador.tfg.com --ip
10.22.122.14
```

Tras la creación del clúster y añadir los nodos que lo conforman al mismo, se puede consultar el fichero **"/etc/ocfs2/cluster.conf"** del Nodo1, que muestra la configuración del mismo. El contenido de este archivo se encuentra en el Anexo XXXII.

El fichero "**cluster.conf**", está presente únicamente en el Nodo1, ya que es el nodo desde dónde se ha creado el clúster, pero debe estar presente en todos los nodos que conforman el clúster OCFS2. Por lo tanto, se hará uso de la utilidad "**scp**" en los nodos restantes del clúster OCFS2 (Nodos 2, 3, 4 y Balanceo), para copiar este fichero de configuración en el directorio **"/etc/ocfs2**". Para ello, se debe crear en los Nodos 2, 3,4 y Balanceo, el directorio **"/etc/ocfs**":

```
$ mkdir -p /etc/ocfs2
```

Una vez creado el directorio anterior, se procederá a copiar el fichero de configuración del clúster OCFS2 del Nodo1, en los nodos restantes (2, 3, 4 y Balanceo) de la siguiente manera:

```
$ scp root@nodo1.tfg.com:/etc/ocfs2/cluster.conf /etc/ocfs2
```

Tras esto, se configurará el clúster OCFS2 en todos los nodos que lo conforman (Nodos 1, 2, 3, 4 y Balanceo). Para la configuración del clúster se hará uso del siguiente comando en todos los nodos del clúster OCFS2:

```
$ /sbin/o2cb.init configure
```

En la ejecución del comando anterior, se pueden dejar por defecto todas las opciones, excepto la tercera opción (**Cluster to start on boot**), ya que se debe indicar el nombre del clúster OCFS2 (En este caso, clusterOCFS2) (Ver ***Ilustración 32***):

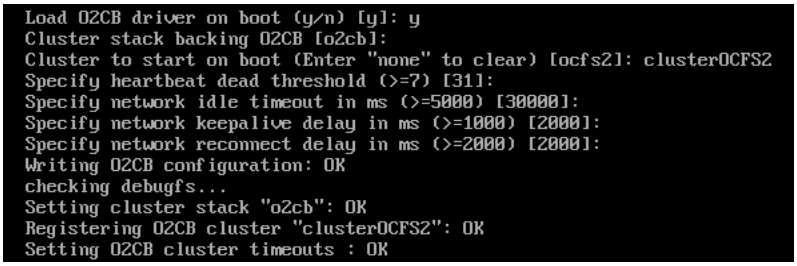

Ilustración 32: Configuración del clúster OCFS2 en cada nodo

Además, se puede comprobar el estado del clúster en cada nodo, ejecutando la orden:

```
$ /sbin/o2cb.init status
```

Tras comprobar que la configuración del clúster es la adecuada, se deben habilitar en el arranque de cada nodo que conforma el clúster OCFS2 (Nodos 1, 2, 3, 4 y Balanceador), los servicios **o2cb** y **ocfs2**:

```
$ systemctl enable o2cb
```

```
$ systemctl enable ocfs2
```

Para finalizar con la configuración del clúster OCFS2, se debe garantizar el correcto funcionamiento del mismo, configurando los ajustes necesarios del kernel. Para ello, se modificará el fichero ***"/etc/sysctl.conf***" de todos los nodos del clúster OCFS2 y se agregará al final del mismo las líneas:

```
kernel.panic=30
kernel.panic_on_oops=1
```

Estas líneas hacen referencia a la entrada en modo "**pánico**" del kernel de Linux en caso de alguna falla en el clúster OCFS2 (reiniciándose por defecto cada nodo en caso de alguna falla).

Tras finalizar con toda la configuración del clúster OCFS2, se creará un volumen físico en alguno de los nodos del clúster haciendo uso del disco importado del servidor iSCSI (**sda**). Tras la creación de este volumen físico, se creará un grupo de volúmenes que alojará el volumen físico anterior. Todo esto, se realiza con el propósito de crear varios volúmenes lógicos utilizando el almacenamiento del grupo de volúmenes recién establecido, para crear en cada uno de ellos un sistema de archivos OCFS2 y garantizar el acceso concurrente a los ficheros de configuración de Apache y de MariaDB desde varios nodos del clúster.

Por lo tanto, desde el Nodo1, se creará un volumen físico sobre el disco de almacenamiento compartido proporcionado por la máquina iscsi.tfg.com (**sda**):

```
$ pvcreate /dev/sda
```

Acto seguido, se creará un grupo de volúmenes llamado "**vg_almacenamiento_compartido**" que albergue el volumen físico recién creado:

```
$ vgcreate vg_almacenamiento_compartido /dev/sda
```

Tras la creación de este grupo de volúmenes que alberga el volumen físico referente al disco compartido iSCSI, se procederá a crear un volumen lógico de 4GB usando el espacio del grupo de volúmenes recién creado, crear un sistema de archivos OCFS2 en él y montarlo en un directorio de los Nodos 1 y 2 en dónde se albergue todos los ficheros de configuración de Apache. Luego, se procederá a crear los links simbólicos correspondientes al directorio de Apache, para ofrecer acceso concurrente a los ficheros de configuración del servidor web, a los Nodos 1 y 2, generando de esta manera un espacio de almacenamiento con alta disponibilidad. Se pueden consultar los detalles para lograr lo anteriormente

descrito en el Anexo XXXIII. Tras todo esto, se procederá de la misma manera, a crear un nuevo volumen lógico de 2GB usando el espacio de almacenamiento del grupo de volúmenes "**vg_almacenamiento_compartido**", con el fin de ofrecer acceso concurrente y alta disponibilidad a los ficheros de configuración de MariaDB en los Nodos 3 y 4, mediante la creación de un sistema de ficheros OCFS2. En el Anexo XXXIV, se pueden consultar todos los pasos para obtener acceso concurrente y alta disponibilidad en los ficheros de configuración de MariaDB de los nodos 3 y 4.

16.22. Prueba del funcionamiento final del clúster que ofrece *Wordpress* a través *Apache* en alta disponibilidad y con balanceo de carga

Finalmente, llegados a este punto, se dispone de un servicio web *Wordpress* basado en *Apache*, que es ofrecido en alta disponibilidad y con balanceo de carga, haciendo uso de un almacenamiento compartido iSCSI síncrono y con alta disponibilidad. Además, las peticiones que se realizan a la base de datos usada por el servicio web de *Wordpress,* están distribuidas de manera balanceada y con alta disponibilidad en los datos de la BD.

Para probar el funcionamiento final de la infraestructura de clúster, y saber que nodo está ofreciendo el servicio web en cada momento, se modificará en el Nodo1 o en el Nodo2, el fichero **"/var/www/html/index.php**" y se agregará justo antes de la etiqueta HTML **"<h1>"** configurada en secciones anteriores, la siguiente línea:

```
$comando = shell_exec('hostname');
```

Tras agregar la línea anterior, se debe modificar la etiqueta HTML "**<h1>"** configurada en apartados anteriores y cambiar su contenido por:

```
echo “<h1 style=’text-align:center;’>SOY EL $comando</h1>”;
```

Tras estos cambios, se accederá desde el navegador del host anfitrión a través de la dirección IP cliente **192.168.122.209**, al servicio web *Wordpress* ofrecido en alta disponibilidad y con balanceo de carga (Ver ***Ilustración 33***):

Ilustración 33: Servicio Wordpress en alta disponibilidad y con balanceo de carga ofrecido por el Nodo1

Como se observa en la imagen anterior, el servicio Wordpress está siendo ofrecido por el Nodo1. Si se realizara alguna modificación en la BD del servicio web de Wordpress, y se recargará la ventana del navegador, entonces el servicio Wordpress pasará a ser ofrecido ahora por el Nodo2, y el cambio realizado en el sitio web (que se registra en la BD) tiene efecto en el Nodo2 y se replica al instante (Ver ***Ilustración 34***):

Ilustración 34: Servicio Wordpress en alta disponibilidad y con balanceo de carga ofrecido por el Nodo2

17. Capítulo XV: Conclusiones y Trabajos futuros

17.1. Conclusiones

Tras el análisis, podemos deducir que la plataforma Oracle Linux en su versión 8.6 es muy similar a CentOS, ya que como se ha podido comprobar en este proyecto, se ha diseñado y desplegado una infraestructura de clúster que proporciona alta disponibilidad y balanceo de carga en sus servicios. De esta manera, en las asignaturas impartidas en la ULPGC tales como: “Infraestructuras Tecnológicas para los Sistemas de Información” y “Virtualización y Procesamiento Distribuido” se podría empezar a utilizar esta distribución de Linux compatible con RHEL en sustitución de CentOS sin ningún inconveniente.

Para alcanzar el objetivo de este proyecto, en primera instancia se tuvo que investigar acerca de las características de la distribución Oracle Linux, el software compatible para la computación en clúster y la plataforma de virtualización recomendable para llevar a cabo el proyecto, concluyendo en que la mayoría de las herramientas utilizadas, están disponibles en la distribución de CentOS y en otras distribuciones de Linux de manera totalmente gratuita.

Por otro lado, la infraestructura de clúster que se ha conseguido diseñar en el transcurso de este trabajo, ofrece un alto rendimiento en sus servicios, ya que se ha conseguido proporcionar alta disponibilidad al servicio web *Wordpress* basado en Apache, a la base de datos utilizada por el servicio web y al almacenamiento compartido iSCSI proporcionado por un servidor externo. Además, se ha logrado ofrecer equilibrio de carga en las peticiones que realizan los potenciales clientes tanto al servicio web como a la base de datos, evitando la sobrecarga en los servidores. Todo ello, se ha conseguido utilizando máquinas virtuales, aprovechando de una mejor manera los recursos informáticos, reduciendo los costes y de la manera más segura posible, filtrando determinadas conexiones a través de un cortafuegos y la arquitectura de seguridad SELinux.

El desarrollo del trabajo no ha sido para nada sencillo, debido a que el diseño y despliegue de la infraestructura creada requiere de una alta complejidad, un dominio avanzado en la administración de sistemas y de red y unas configuraciones avanzadas en las diferentes herramientas software instaladas para lograr el funcionamiento requerido.

Por todo ello, se considera que este proyecto se ha concluido de una manera satisfactoria, ya que se ha logrado el objetivo preestablecido y se ha descubierto una muy buena alternativa a la distribución de Linux CentOS, para seguir llevando a cabo el transcurso de algunas asignaturas impartidas en la ULPGC.

17.2. Trabajos futuros

Como ya se ha mencionado anteriormente, se ha diseñado a través de la distribución de Oracle Linux, una infraestructura de clúster que ofrece alta disponibilidad y balanceo de carga a sus servicios de una manera bastante acertada. De igual manera, existen opciones para mejorar el desempeño de esta infraestructura, tales como:

- Configuración del servidor web Apache y el servicio HAProxy, mediante un certificado SSL autofirmado que permita un tráfico web más seguro a través del protocolo HTTPS.
- Configuración de un nuevo balanceador de carga de respaldo, ya que en la infraestructura diseñada se dispone de un único balanceador, que si fallara se perdería la alta disponibilidad y el equilibrio de carga ofrecido. Por ello, sería muy buena idea configurar un nuevo nodo Balanceador de carga para generar redundancia.

- Configuración de un nombre de dominio en el host anfitrión, para poder acceder al servicio web Wordpress ofrecido, mediante un nombre dominio en lugar de utilizar la dirección IP cliente.

- Estudiar en profundidad las posibilidades que tiene el sistema de archivos distribuido OCFS2 para trabajar con SELinux en modo "*enforcing*" en todos los nodos de la infraestructura.

18. Anexos

Anexo I: Pasos para la instalación de Oracle Linux 8.6

Una vez arrancado el sistema anfitrión con la unidad USB *booteable* toma el control el instalador del sistema operativo correspondiente, en dónde hay que seguir una serie de pasos para poder empezar a usar el S.O:

Paso 1: Selección del idioma

En este paso se debe escoger el idioma a utilizar durante el proceso de instalación, que en este caso será el idioma Español (Ver ***Ilustración 35***):

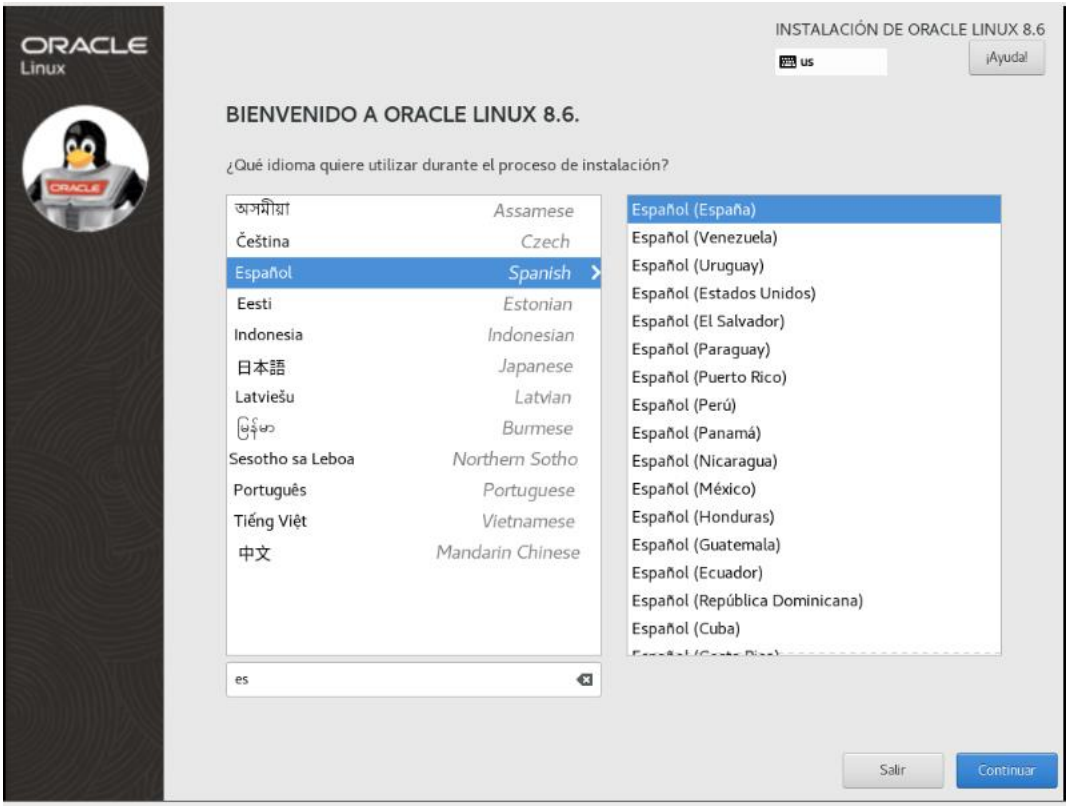

Ilustración 35: Selección del idioma para la instalación

Paso 2: Selección de la configuración del sistema, software, ajustes de usuario y regionalización

En este paso, se deben ajustar determinados parámetros que tendrá el sistema operativo a instalar, como, por ejemplo: el particionado (partición raíz ("/"), partición de arranque ("*boot*") y partición *home ("home")*), el software a instalar, la conexión de red, idioma, la fecha y la hora, distribución del teclado, contraseña del usuario administrador (root), creación de un nuevo usuario, etc. (Ver ***Ilustración 36***).

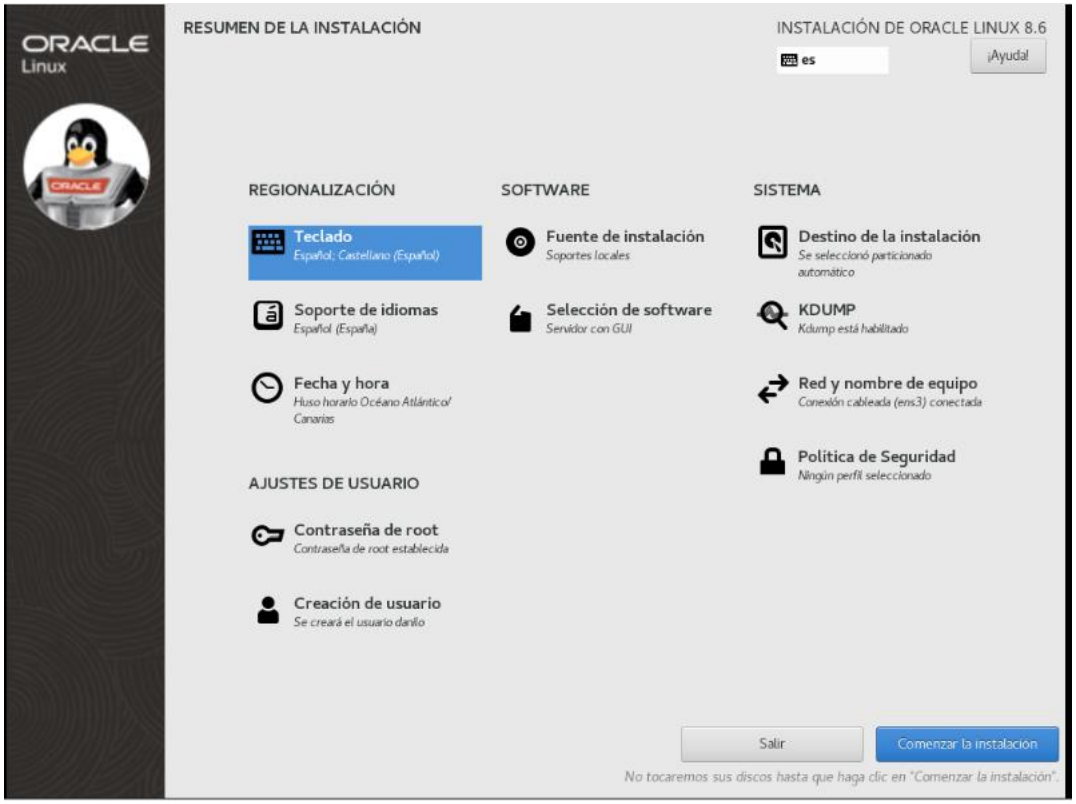

Ilustración 36: Ajustes de la instalación del Sistema operativo

Cabe mencionar, que se ha optado por instalar un entorno gráfico para un mejor manejo del sistema anfitrión. Por lo que en la selección del software se ha seleccionado la opción de "*Servidor con GUI*" para instalar el típico entorno de escritorio GNOME.

Paso 3: Instalación

Una vez llegados a este paso, comenzará la instalación de la distribución Oracle Linux 8.6 y todo el software necesario y el que se ha seleccionado en el paso anterior (Ver ***Ilustración 37***).

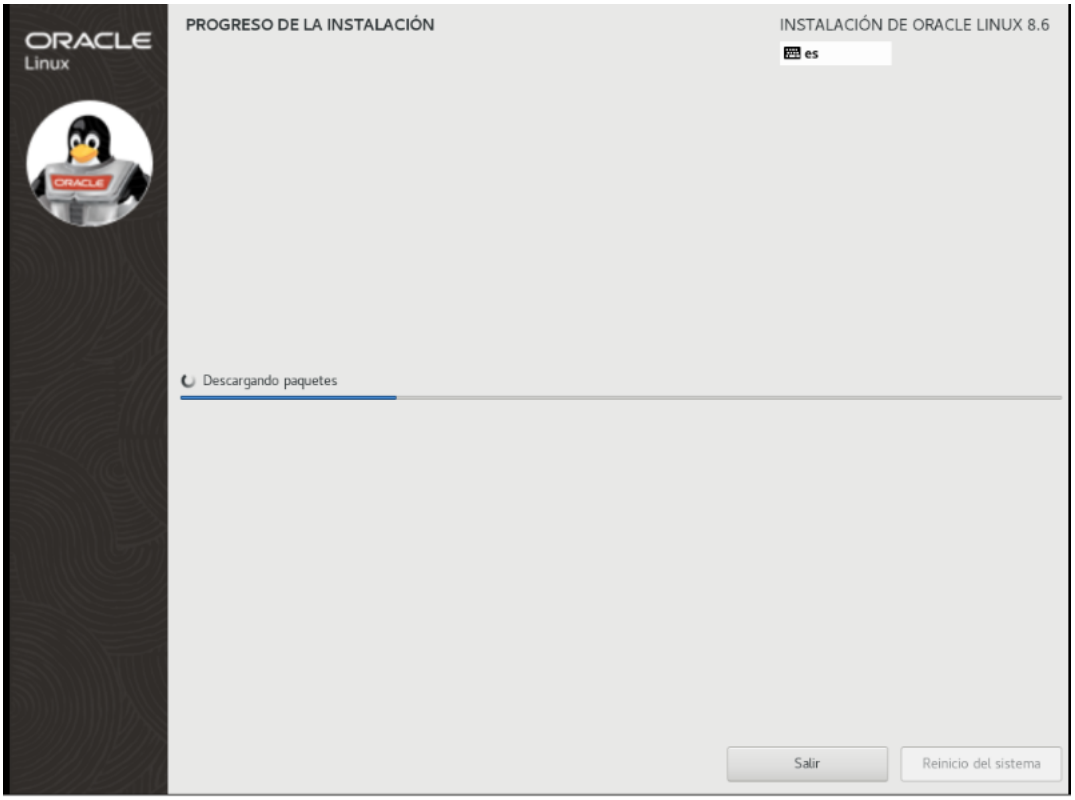

Ilustración 37: Progreso de la instalación de Oracle Linux 8.6

Una vez terminada la instalación, solo queda reiniciar el sistema para poder empezar a usar *Oracle Linux (Ver* **Ilustración 38***)*.

Ilustración 38: Fin de la instalación de Oracle Linux y reinicio del Sistema Host Anfitrión

Tras el reinicio del sistema e iniciar sesión con alguno de los usuarios creados en el proceso de instalación, ya se debe tener en funcionamiento *Oracle Linux 8.6* con su entorno de escritorio gráfico GNOME (Ver ***Ilustración 39***).

Ilustración 39: Entorno gráfico de escritorio GNOME

Anexo II: Detalles de los requerimientos del sistema host y del sistema operativo invitado

Requisitos y recomendaciones del sistema anfitrión:

Todos los requisitos y recomendaciones que se exponen a continuación están accesibles en [32, Cap. 1].

- **Anfitrión de metal desnudo:** KVM solo es compatible cuando se ejecuta en lo que se denomina un *host bare metal*, ya que los escenarios de virtualización anidados no son compatibles con KVM.

- **CPU:** Con respecto a la CPU del sistema host, se debe tener habilitadas las funciones de virtualización Intel (*VT-x*) o AMD (*AMD-V*), según corresponda. Para verificar que las funciones de virtualización están habilitadas en el host anfitrión, podemos observar el contenido del fichero /proc/cpuinfo y buscar en él los flags *vmx* (caso de Intel) o *svm* (caso de AMD) (Ver ***Ilustración 40***).

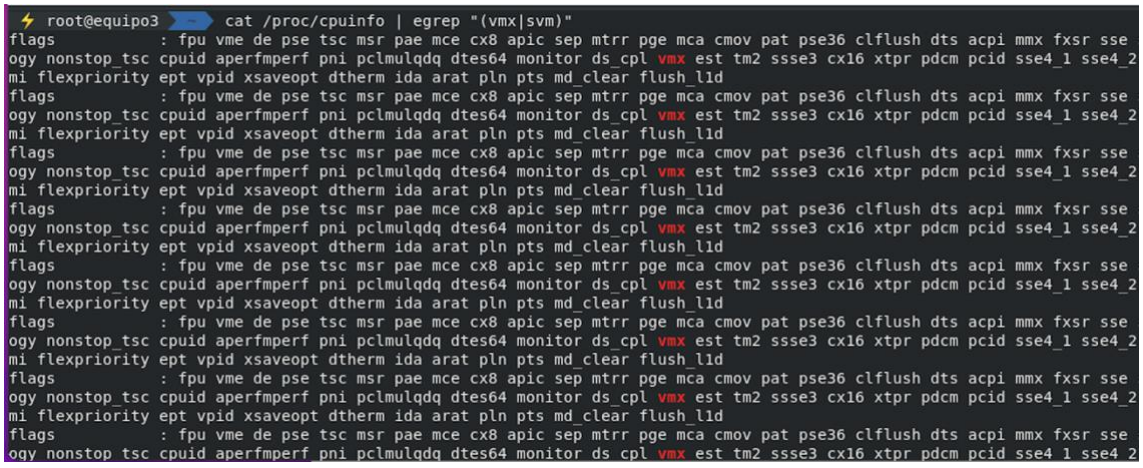

Ilustración 40: Funciones de Virtualización para Intel habilitadas

Otra manera de comprobar si el host puede admitir la virtualización es haciendo uso del comando "*lscpu*" y estudiar los términos que aparecen en la

sección "*Indicadores*", donde debe aparecer el término *vmx* (Intel) o bien *svm* (AMD), según corresponda. Si ninguno de estos dos términos (*vmx* o *svm*) aparecen, entonces el host no permite la virtualización y se tendría que habilitar en la BIOS del sistema (Ver ***Ilustración 41***).

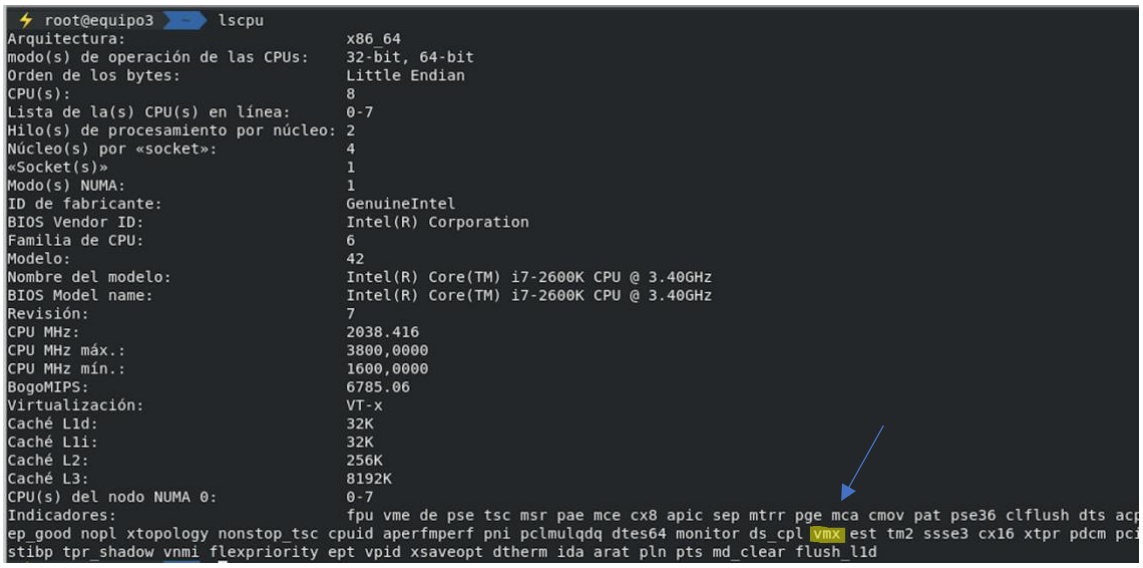

Ilustración 41: Funciones Virtualización para Intel con el comando lscpu

Además, como regla general, se puede comenzar a trabajar con las siguientes proporciones de CPU virtual a CPU host:

- **1:1** a **2:1** logra un buen rendimiento de las máquinas virtuales.
- **3:1** podría generar algunos inconvenientes en el rendimiento de la máquina virtual.
- **4:1** o **superior**, podría causar problemas graves en el rendimiento de la máquina virtual.

Estas proporciones se determinan mediante pruebas de rendimiento tanto en la máquina virtual como en el sistema host y hay que tener en cuenta factores tales como: el volumen de tareas a ejecutar, la velocidad deseada de procesamiento de esas tareas y las tareas que realizan las máquinas virtuales.

- **Memoria:** La reserva de 3GB de memoria para el host anfitrión es un buen punto de partida, pero los requerimientos de memoria para el sistema operativo del host (Oracle Linux) aumentan con la cantidad de memoria

  física que se tenga. Por ejemplo, para un sistema con 1TB de memoria, se recomienda tener al menos 20GB disponibles de memoria para el S.O host. En el caso de que las máquinas virtuales excedan la memoria RAM física disponible, el impacto en el rendimiento sería considerable.
  Se puede hacer uso del comando "*free"* para observar la memoria RAM disponible, libre, usada, etc. en el sistema host (Ver ***Ilustración 42***).

Ilustración 42: Información de la memoria del sistema

Almacenamiento: Como mínimo se debe tener un espacio libre en disco de 6GB para el sistema operativo del host. Además, también se debe atender alrededor de 6GB de espacio de almacenamiento por máquina virtual, considerando el propósito de la máquina virtual y ajustar en consecuencia.
Para obtener información acerca del espacio total, ocupado y libre en el sistema, se puede hacer uso del comando "*df*".

Requisitos del sistema operativo invitado:

En la siguiente tabla se describen los sistemas operativos invitados de Linux que se pueden utilizar en una instancia de KVM.

Sistema Operativo Linux	Arquitectura de 32 bits	Arquitectura de 64 bits
oracle linux 6	Sí*	Sí
oracle linux 7	No disponible	Sí
oracle linux 8	No disponible	Sí
Red Hat Enterprise Linux 6	Sí*	Sí
Red Hat Enterprise Linux 7	No disponible	Sí
Red Hat Enterprise Linux 8	No disponible	Sí
CentOS 6	Sí*	Sí
CentOS 7	No disponible	Sí
CentOS 8	No disponible	Sí
SUSE Linux Enterprise Server 12 SP5	No disponible	Sí
SUSE Linux Enterprise Server 15 SP1	No disponible	Sí
Ubuntu 16.04	No disponible	Sí
Ubuntu 18.04	No disponible	Sí
Ubuntu 20.04	No disponible	Sí

Tabla 4: Sistemas operativos invitados Linux soportados por KVM

Por otro lado, los sistemas operativos invitados de *Microsoft Windows* compatibles con la plataforma KVM son:

Sistema operativo Microsoft Windows	Arquitectura de 32 bits	Arquitectura de 64 bits
Servidor Microsoft Windows 2019	No disponible	Sí
Servidor Microsoft Windows 2016	No disponible	Sí
Servidor Microsoft Windows 2012 R2	No disponible	Sí
Servidor de Microsoft Windows 2012	No disponible	Sí
Servidor Microsoft Windows 2008 R2 SP1	No disponible	Sí
Servidor de Microsoft Windows 2008 SP1	Sí	Sí
microsoft windows 10	Sí	Sí
microsoft windows 8.1	Sí	Sí
microsoft windows 8	Sí	Sí
Microsoft Windows 7 SP1	Sí	Sí

Tabla 5: Sistemas operativos invitados Windows soportados por KVM

Por último, se puede utilizar como sistema operativo invitado cuando se instala dentro de una instancia independiente de KVM, *Oracle Solaris*. Además, *Oracle Solaris 11.4.33*, es la versión mínima compatible con el controlador *VirtIO*.

Anexo III: Información de los grupos de paquetes de virtualización instalados para KVM y sus paquetes

Se puede usar la orden "*dnf groupinfo*" para obtener la información necesaria de cada grupo de paquetes y ver de qué paquetes está compuesto el grupo (Ver ***Ilustración 43***):

```
root@equipo3 ~ dnf groupinfo "Virtualization Hypervisor"
Última comprobación de caducidad de metadatos hecha hace 1:40:48, el vie 08 abr 2022 12:22:33 WEST.

Grupo: Hipervisor de virtualización
 Descripción: La instalación de host de virtualización más pequeña posible.
 Paquetes obligatorios:
   libvirt
   qemu-kvm
root@equipo3 ~ dnf groupinfo "Virtualization Client"
Última comprobación de caducidad de metadatos hecha hace 1:40:54, el vie 08 abr 2022 12:22:33 WEST.

Grupo: Cliente de virtualización
 Descripción: Clientes para instalar y administrar instancias de virtualización.
 Paquetes obligatorios:
   gnome-boxes
   virt-install
   virt-manager
   virt-viewer
 Paquetes predeterminados:
   virt-top
 Paquetes opcionales:
   libguestfs-inspect-icons
   libguestfs-tools
   libguestfs-tools-c
root@equipo3 ~ dnf groupinfo "Virtualization Platform"
Última comprobación de caducidad de metadatos hecha hace 1:41:00, el vie 08 abr 2022 12:22:33 WEST.

Grupo: Plataforma de virtualización
 Descripción: Proporciona una interfaz para acceder y controlar huéspedes y contenedores virtualizados.
 Paquetes obligatorios:
   libvirt
   libvirt-client
 Paquetes opcionales:
   fence-virtd-libvirt
   fence-virtd-multicast
   fence-virtd-serial
   perl-Sys-Virt
root@equipo3 ~ dnf groupinfo "Virtualization Tools"
Última comprobación de caducidad de metadatos hecha hace 1:41:06, el vie 08 abr 2022 12:22:33 WEST.

Grupo: Herramientas de virtualización
 Descripción: Herramientas para gestión de imagen virtual desconectada.
 Paquetes predeterminados:
   libguestfs
 Paquetes opcionales:
   libguestfs-inspect-icons
   libguestfs-java
   libguestfs-tools
   libguestfs-tools-c
root@equipo3 ~
```

Ilustración 43: Información de los grupos de paquetes de virtualización instalados para KVM

Anexo IV: Creación de una máquina virtual mediante *virt-manager*

Crear una máquina virtual haciendo uso de la herramienta gráfica *virt-manager* es bastante sencillo. En este caso, se creará una máquina virtual que ejecuta un sistema operativo Oracle Linux en su versión 8.6. Por lo tanto, antes que nada, se debe descargar la imagen ISO de Oracle Linux 8.6 desde internet, como se ha mencionado anteriormente en esta memoria. Tras esto, se procederá a usar *virt-manager* para la creación de la máquina virtual:

Paso 1: Abrir la utilidad virt-manager

Como se ha mencionado con anterioridad, se procederá a utilizar *virt-manager* para la creación de una máquina virtual, por lo que debemos iniciar esta utilidad. Para ello hay dos formas, o bien haciendo clic en la aplicación "*Gestión de máquinas virtuales*" que se encuentra en las Herramientas del sistema o ejecutando el comando *"virt-manager"* como se muestra en la ***Ilustración 44*** para acceder a la interfaz-grafica de KVM:

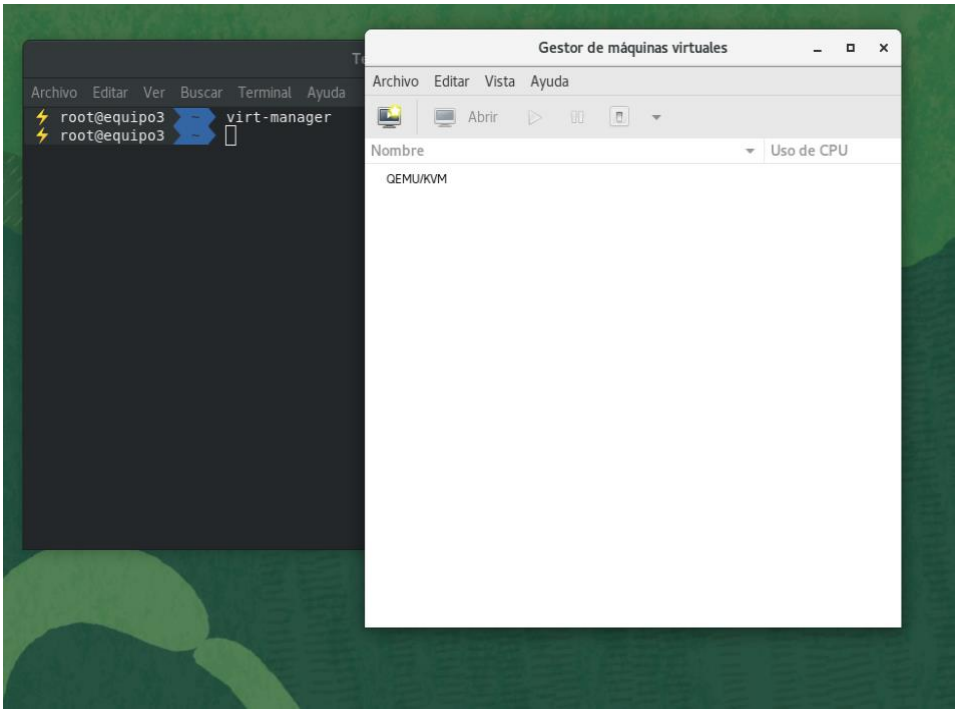

Ilustración 44: Utilidad virt-manager

Paso 2: Medio de instalación de la MV

Tras hacer clic en el icono  para crear una nueva máquina virtual, en este paso se seleccionará el medio de instalación, que en este caso será local, ya que se trata de una imagen ISO que tenemos descargada previamente (Ver ***Ilustración 45***).

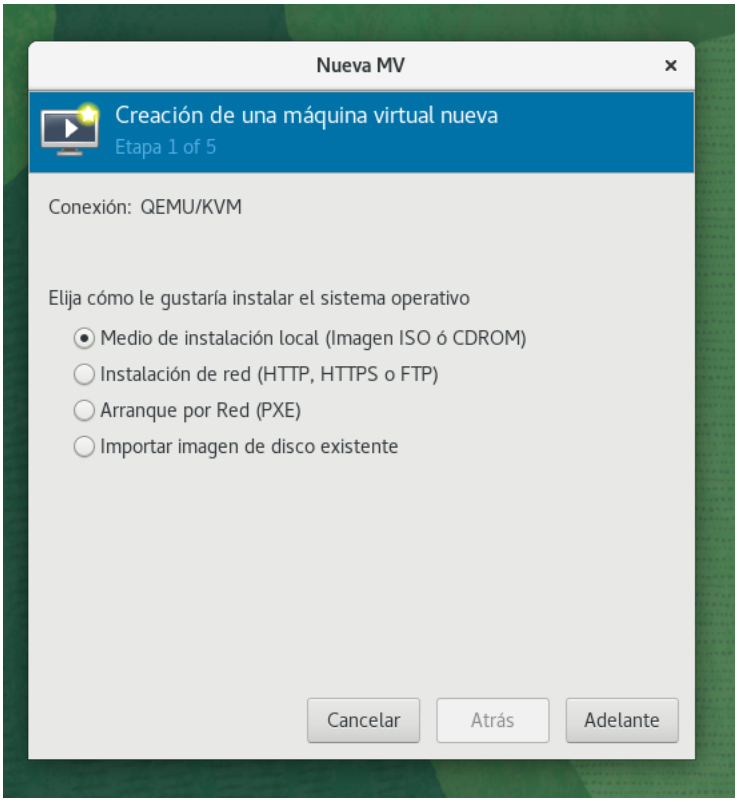

Ilustración 45: Medio de instalación de la máquina virtual

Paso 3: Selección de la imagen ISO

En este paso se seleccionará la imagen ISO de Oracle Linux que se ha descargado previamente (Ver ***Ilustración 46***).

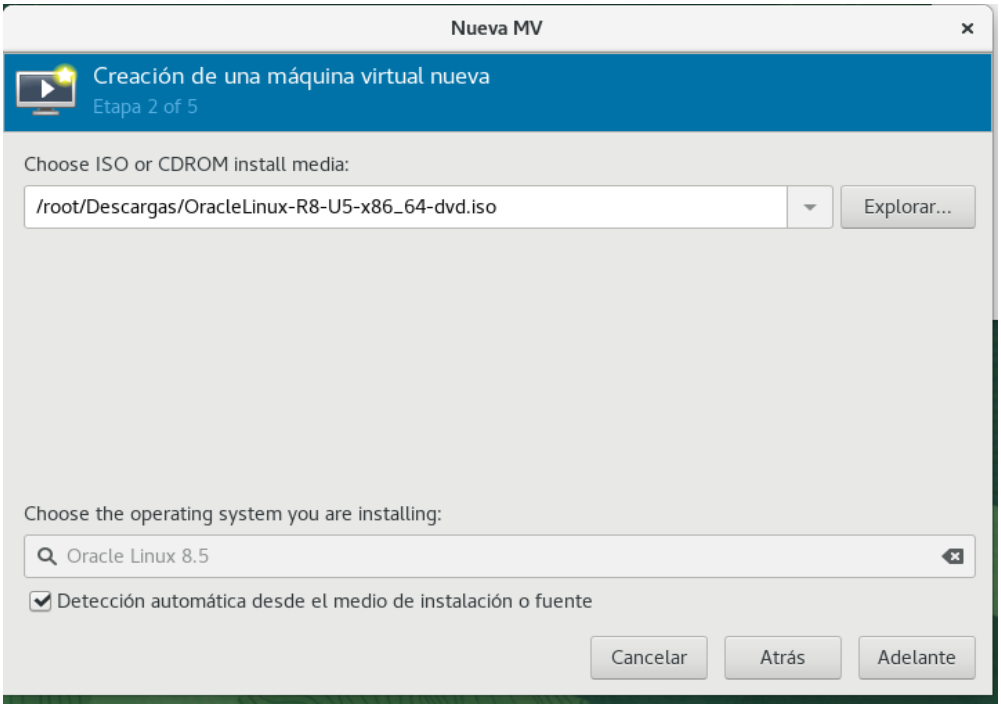

Ilustración 46: Selección de la imagen ISO

Paso 4: Memoria RAM y CPU

En este paso se asignará la memoria RAM (1GB en este caso) y el número de CPU's (1 en este caso), como se muestra en la ***Ilustración 47***:

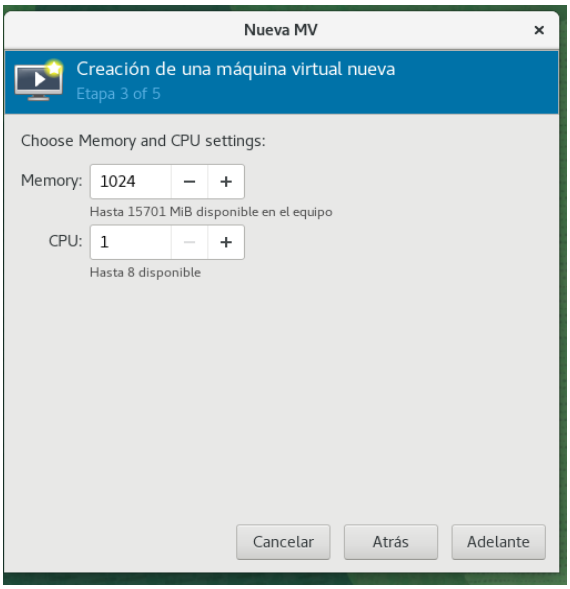

Ilustración 47: Memoria y CPU de la máquina virtual

Paso 5: Espacio en disco

En este paso se creará una imagen de disco para la máquina virtual (de 10GB en este caso):

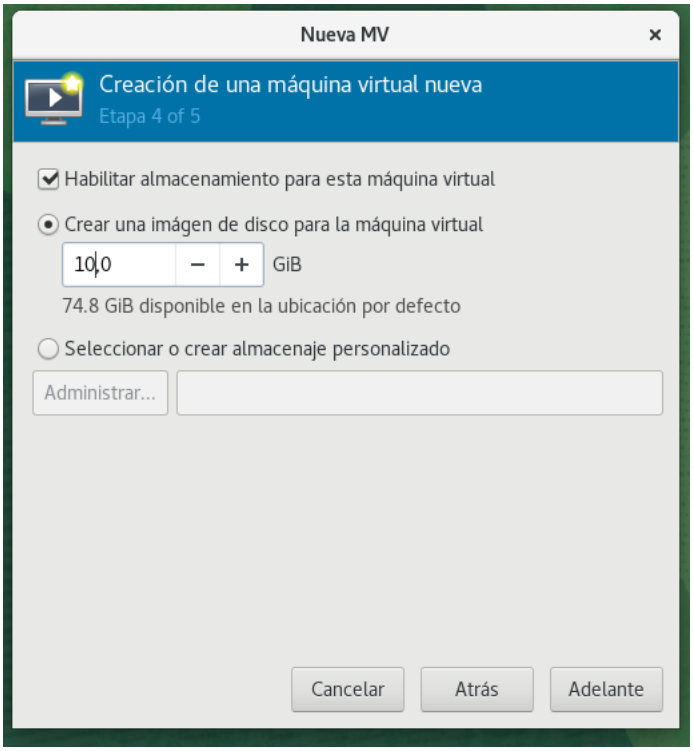

Ilustración 48: Imagen de disco de la MV

Paso 6: Nombre de la MV y Red

En este paso se le asignará el nombre a la máquina virtual (PruebaVirtManager) y se seleccionará la red a utilizar (Red virtual de tipo NAT).

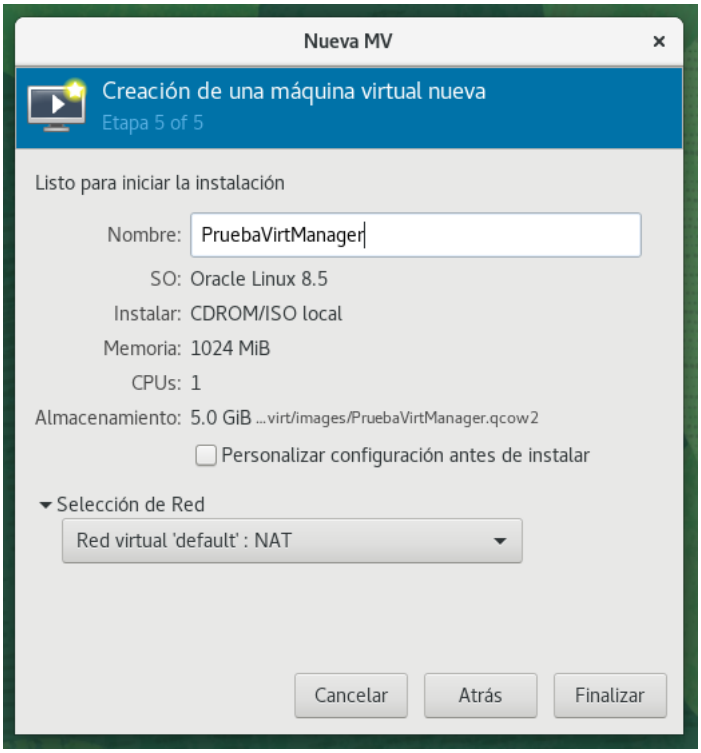

Ilustración 49: Nombre y red de la máquina virtual

Paso 7: Comprobaciones

Una vez proporcionado todos los parámetros anteriores, el instalador del S.O correspondiente tomará el control.

Tras finalizar la instalación, ya tendremos la máquina virtual funcionando con una instalación mínima de Oracle Linux, por lo que se procederá a comprobar la conectividad:

Conectividad de la máquina virtual con el host anfitrión:

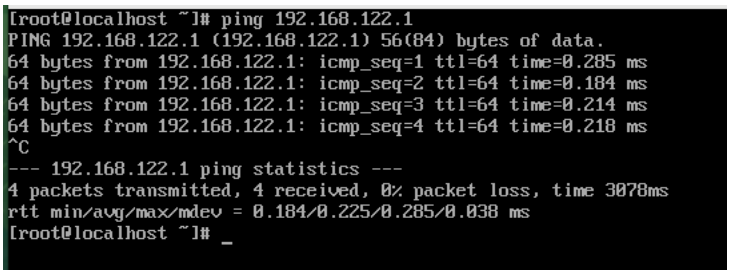

Ilustración 50: Comprobación de la conectividad de la MV con el host

Conectividad de la máquina virtual con el exterior:

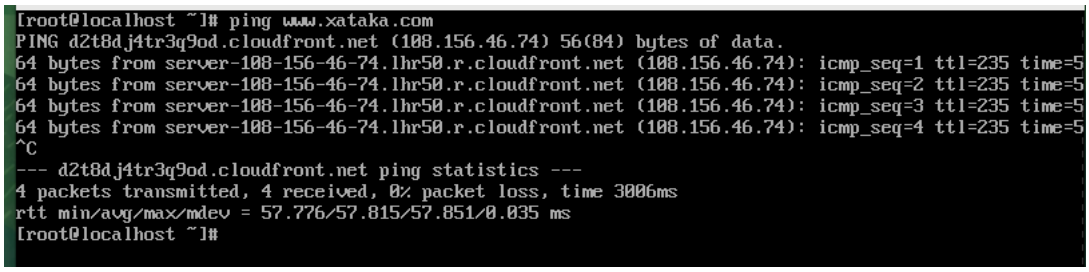

Ilustración 51: Comprobación de conectividad al exterior

Anexo V: Pasos para realizar una copia de seguridad manual de una máquina virtual

Como se ha comentado el Capítulo XI, para realizar una copia de seguridad manualmente de una máquina virtual, se deben tener en cuenta dos ficheros, el fichero XML de configuración de la máquina virtual y su imagen de disco. Por lo tanto, lo primero que se debe hacer es localizar estos ficheros en el sistema haciendo uso de la orden "*find*" como se muestra en la ***Ilustración 52***:

Ilustración 52: Ficheros necesarios para realizar una copia de seguridad de una máquina virtual de KVM

Como se puede observar en la imagen anterior, se ha buscado con el comando "*find*" bajo el directorio raíz ("/") los ficheros que tengan como nombre el dominio de la máquina virtual de KVM a copiar, que en este caso recibe el nombre de "*PruebaVirtManager*". La salida de la ejecución del comando anterior arroja los dos ficheros mencionados anteriormente, el fichero de configuración XML y la imagen de disco (fichero con extensión qcow2).

Una vez localizados estos dos ficheros, se realizará una copia de seguridad de los mismos. Teniendo en cuenta que la máquina virtual a copiar tiene el nombre de "PruebaVirtManager", una forma de realizar la copia de seguridad de estos ficheros puede ser esta (Ver ***Ilustraciones 53 y 54***):

```
root@equipo3 ~ cp /etc/libvirt/qemu/PruebaVirtManager.xml /etc/libvirt/qemu/Prueba_copiando_ficheros.xml
'/etc/libvirt/qemu/PruebaVirtManager.xml' -> '/etc/libvirt/qemu/Prueba_copiando_ficheros.xml'
```

Ilustración 53: Copia de seguridad del fichero de configuración XML de la máquina virtual

```
root@equipo3 ~ cp /var/lib/libvirt/images/PruebaVirtManager.qcow2 /var/lib/libvirt/images/Prueba_copiando_ficheros.qcow2
'/var/lib/libvirt/images/PruebaVirtManager.qcow2' -> '/var/lib/libvirt/images/Prueba_copiando_ficheros.qcow2'
```

Ilustración 54: Copia de seguridad de la imagen de disco de la máquina virtual

Como se observa, se ha hecho uso del comando "*cp*" para realizar la copia de seguridad y se ha realizado una copia denominada "Prueba_copiando_ficheros.xml" del fichero XML de la máquina virtual llamada "PruebaVirtManager". De la misma manera, se ha realizado una copia de seguridad de la imagen de disco de la máquina "PruebaVirtManager" denominada "Prueba_copiando_ficheros.qcow2". Es muy importante que antes de realizar una copia de la imagen de disco de una máquina virtual, nos aseguremos que la misma esté apagada, ya que si no fuera así se podrían obtener problemas graves al realizar la copia.

Tras realizar las copias de seguridad anteriores, se deben efectuar una serie de modificaciones en el fichero de copia denominado

"Prueba_copiando_ficheros.xml". Para ello, se debe abrir este fichero de configuración con un editor de texto (gedit, vi, nano, vim …), y editar en primer lugar el nombre a la máquina virtual, modificando la etiqueta **<name>** del fichero XML (En este caso se le asignará como nombre a la nueva máquina virtual "Prueba_copiando_ficheros").

A continuación, debemos de asignarle un nuevo UUID (*Universally Unique IDentifier*) a la nueva máquina virtual que vamos a crear como copia de seguridad. Para ello se tienen dos opciones, la primera es, generar un nuevo UUID a través de la orden "*uuidgen*" propia de Linux y asignárselo a la etiqueta **<uuid>** del fichero XML, y la segunda opción consiste en eliminar directamente la etiqueta **<uuid>** y su contenido del fichero XML.

En tercer lugar, debemos generar una nueva dirección MAC (Media Access Control) para asignársela a la nueva máquina virtual que vamos a crear. Se puede generar una nueva dirección MAC a través del generador aleatorio de direcciones MAC disponible en [36]. Tras obtener esta nueva dirección MAC, debemos de modificar la etiqueta "**<mac address/>"** del fichero XML y asignarle la nueva dirección física obtenida.

Por último, se debe de modificar la etiqueta **<source file="ruta"/>** y asignarle la ruta del fichero en donde se ha copiado la imagen del disco de la máquina virtual original (En este caso en /var/lib/libvirt/images/Prueba_copiando_ficheros.qcow2).

Tras realizar todos estos cambios, ya estamos preparados para crear una nueva máquina virtual haciendo uso del fichero XML copiado y modificado anteriormente y de la utilidad de línea de comandos llamada "*virsh*". Por lo que se procederá con la creación de esta nueva máquina virtual de resguardo denominada "Prueba_copiando_ficheros" mediante la utilidad de KVM de línea de comandos "*virsh*":

```
$ virsh define /etc/libvirt/qemu/Prueba_copiando_ficheros.xml
```

Una vez ejecutado el comando anterior, podemos corroborar que la máquina virtual creada como copia de seguridad anteriormente se ha creado con éxito haciendo uso del comando "*virsh list --all*" que lista todas las máquinas virtuales de KVM disponibles (Ver ***Ilustración 55***):

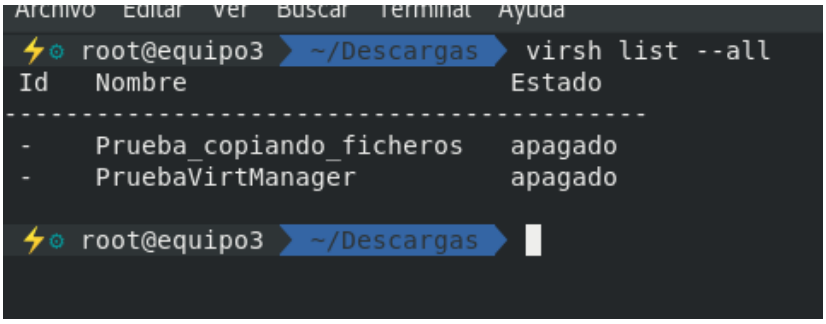

Ilustración 55: Listado de máquinas virtuales disponibles

Finalmente, se debe arrancar la máquina virtual recién creada, bien sea, desde el "*virt manager*" o con la orden "*virsh start Prueba_copiando_ficheros*". Una vez arrancada la nueva máquina virtual debemos de hacer comprobaciones de red, configuraciones de IP si fuera necesario, etc.

Anexo VI: Pasos para realizar una clonación mediante virt-manager

Realizar una clonación mediante *virt-manager* es una tarea muy sencilla, ya que tan sólo hay que hacer clic derecho sobre la máquina virtual que se quiera clonar (en este caso PruebaVirtManager) y se desplegará la siguiente ventana (Ver ***Ilustración 56***) en donde se le ha de indicar el nombre de la nueva máquina virtual a crear:

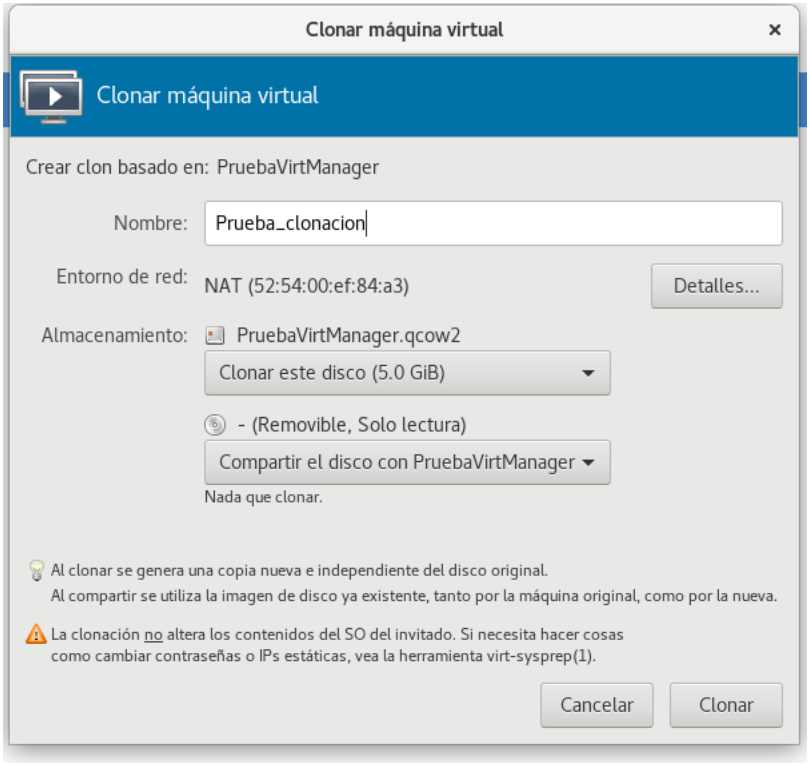

Ilustración 56: Clonación con virt-manager de una máquina virtual

Acto seguido, debemos de asignarle una nueva dirección MAC a la nueva máquina clonada (haciendo clic en “Detalles”), como se muestra en la ***Ilustración 57***.

Nota: La nueva dirección MAC, la podemos generar haciendo uso del generador de direcciones MAC aleatorias usado anteriormente en el Anexo V.

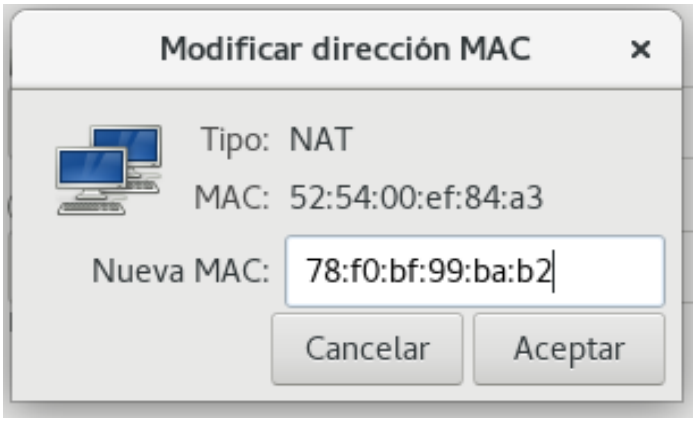

Ilustración 57: Asignación de una nueva dirección MAC pata la máquina virtual clona

Tras realizar los pasos anteriores, ya tendríamos nuestra nueva máquina virtual clonada con *virt-manager*. Al ponerla en funcionamiento debemos de hacer comprobaciones de red, configuraciones de IP (si fuera necesario), etc.

Anexo VII: Pasos para la creación de un volumen en un contenedor de almacenamiento y asociación a una máquina virtual mediante *virt-manager* y *virsh*

Con Virt-manager

Para crear un volumen haciendo uso de *virt-manager* se tendrá que navegar a "*Editar-Detalles de la conexión-Almacenamiento"* y seleccionar el contenedor por defecto llamado "*default*". Una vez seleccionado el contenedor "*default*" se debe hacer clic en el icono  que está sobre la lista de volúmenes. Al crear un volumen se debe especificar: nombre, formato y capacidad, como se muestra en la ***Ilustración 58***:

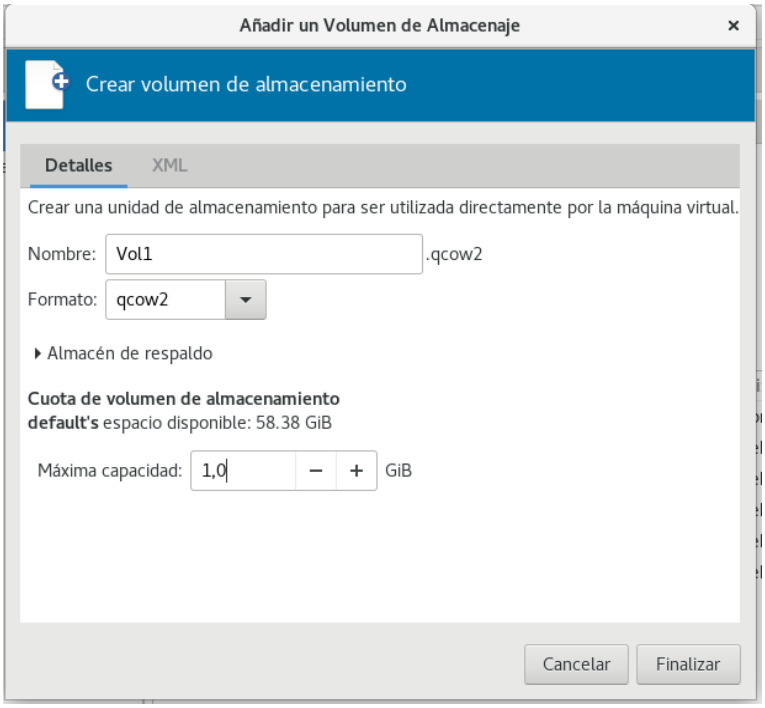

Ilustración 58: Creación de un volumen de almacenamiento en el contenedor default

Una vez creado el volumen, se puede corroborar su existencia a través de la lista de volúmenes que proporciona la interfaz gráfica de virt-manager como se muestra en la ***Ilustración 59***:

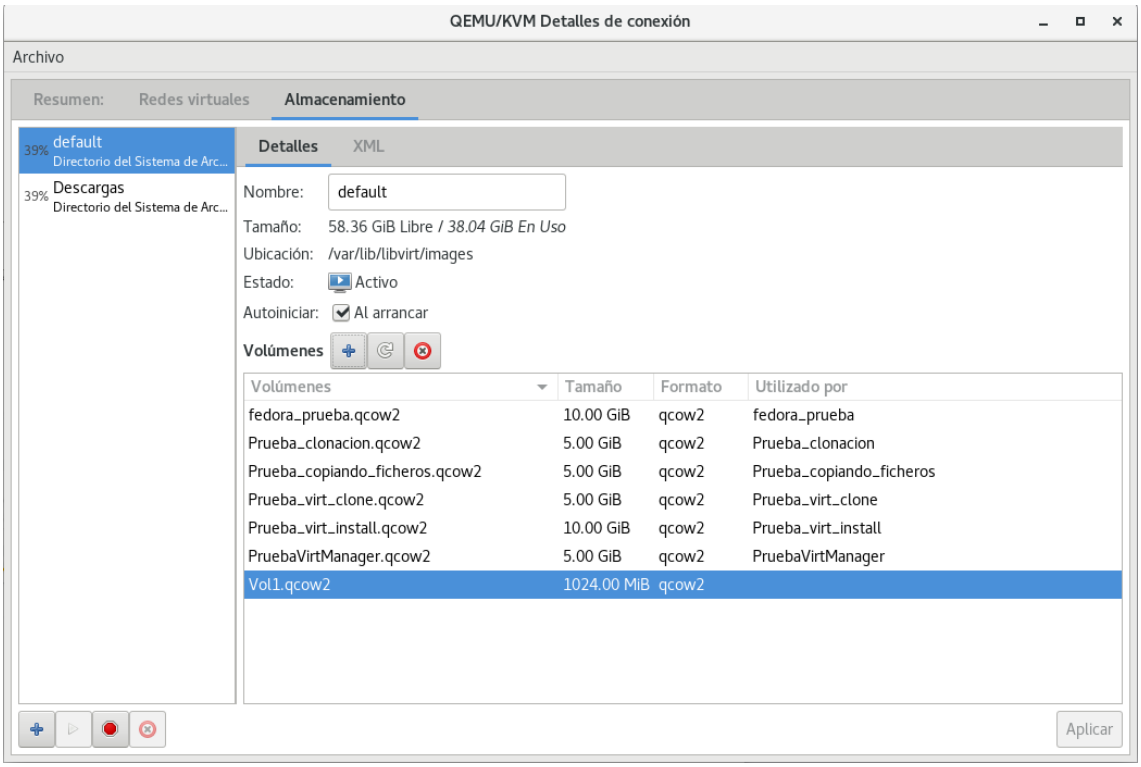

Ilustración 59: Existencia del volumen recién creado dentro del contenedor por defecto

También, se puede corroborar su existencia mostrando la lista de volúmenes del contenedor "*default*" con la utilidad *virsh (*Ver ***Ilustración 60****):*

```
root@equipo3 .../sysconfig/network-scripts virsh vol-list default
Nombre                          Ruta
-----------------------------------------------------------------------------------------
fedora_prueba.qcow2             /var/lib/libvirt/images/fedora_prueba.qcow2
Prueba_clonacion.qcow2          /var/lib/libvirt/images/Prueba_clonacion.qcow2
Prueba_copiando_ficheros.qcow2  /var/lib/libvirt/images/Prueba_copiando_ficheros.qcow2
Prueba_virt_clone.qcow2         /var/lib/libvirt/images/Prueba_virt_clone.qcow2
Prueba_virt_install.qcow2       /var/lib/libvirt/images/Prueba_virt_install.qcow2
PruebaVirtManager.qcow2         /var/lib/libvirt/images/PruebaVirtManager.qcow2
Vol1.qcow2                      /var/lib/libvirt/images/Vol1.qcow2

root@equipo3 .../sysconfig/network-scripts
```

Ilustración 60: Lista de volúmenes del contenedor default con virsh

Tras comprobar la creación exitosa del volumen en el contenedor por defecto, para poder utilizarlo, se asignará el volumen a una máquina virtual. Esta tarea es muy sencilla haciendo uso de *virt-manager*, ya que tan solo se debe seleccionar la máquina virtual (en este caso la máquina virtual llamada PruebaVirtManager) y navegar a "*Editar-Detalles de la máquina virtual-Mostrar detalles de hardware virtual-Agregar Hardware*". Para agregar el volumen recientemente creado, se seleccionará el apartado de almacenamiento del panel de hardware virtual de la MV y se debe seleccionar el volumen anteriormente creado como se muestra en la ***Ilustración 61***:

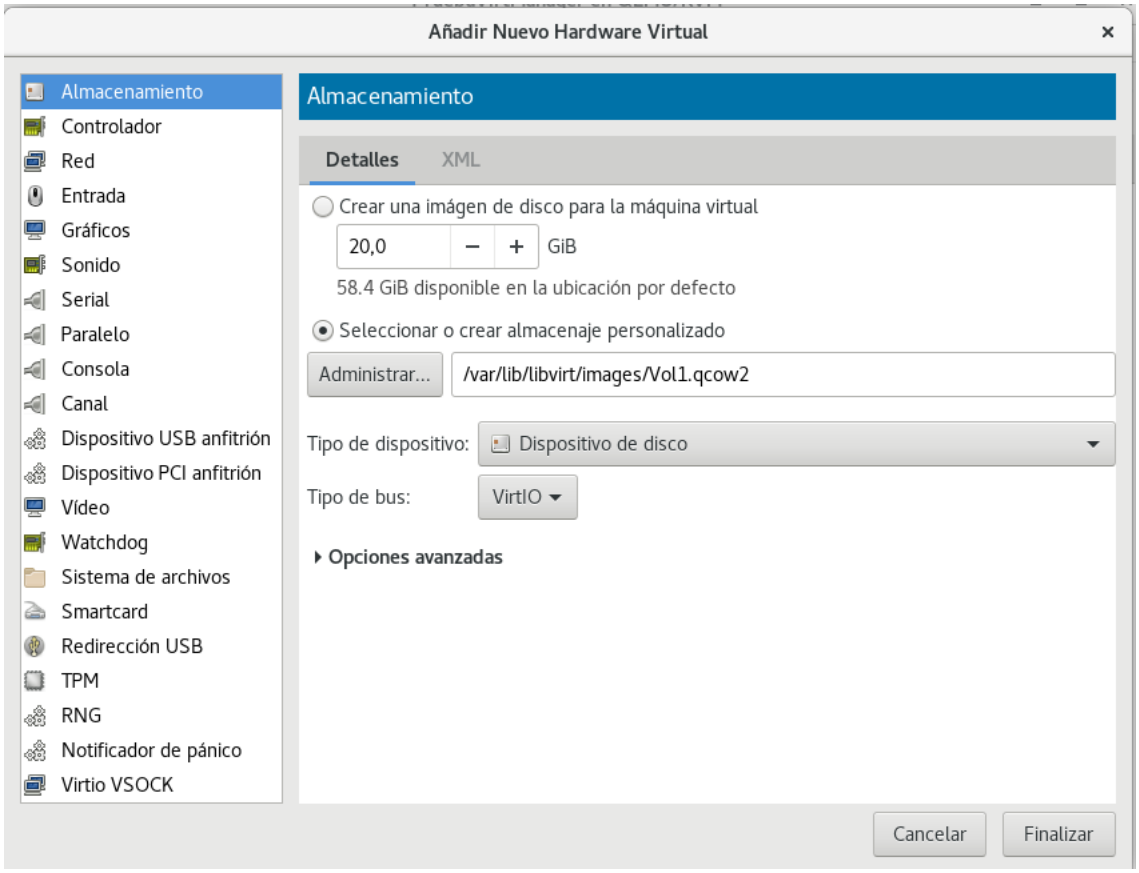

Ilustración 61: Asociación del volumen recién creado a una máquina virtual mediante virt-manager

Tras hacer clic en el botón "*Finalizar*", se habrá asociado el volumen a la máquina virtual "*PruebaVirtPanager*". De igual manera, se puede comprobar que esta máquina virtual dispone de este volumen mostrando sus detalles de

hardware navegando a "*Editar-Detalles de la máquina virtual-Mostrar detalles de hardware virtual*" como se muestra en la ***Ilustración 62***:

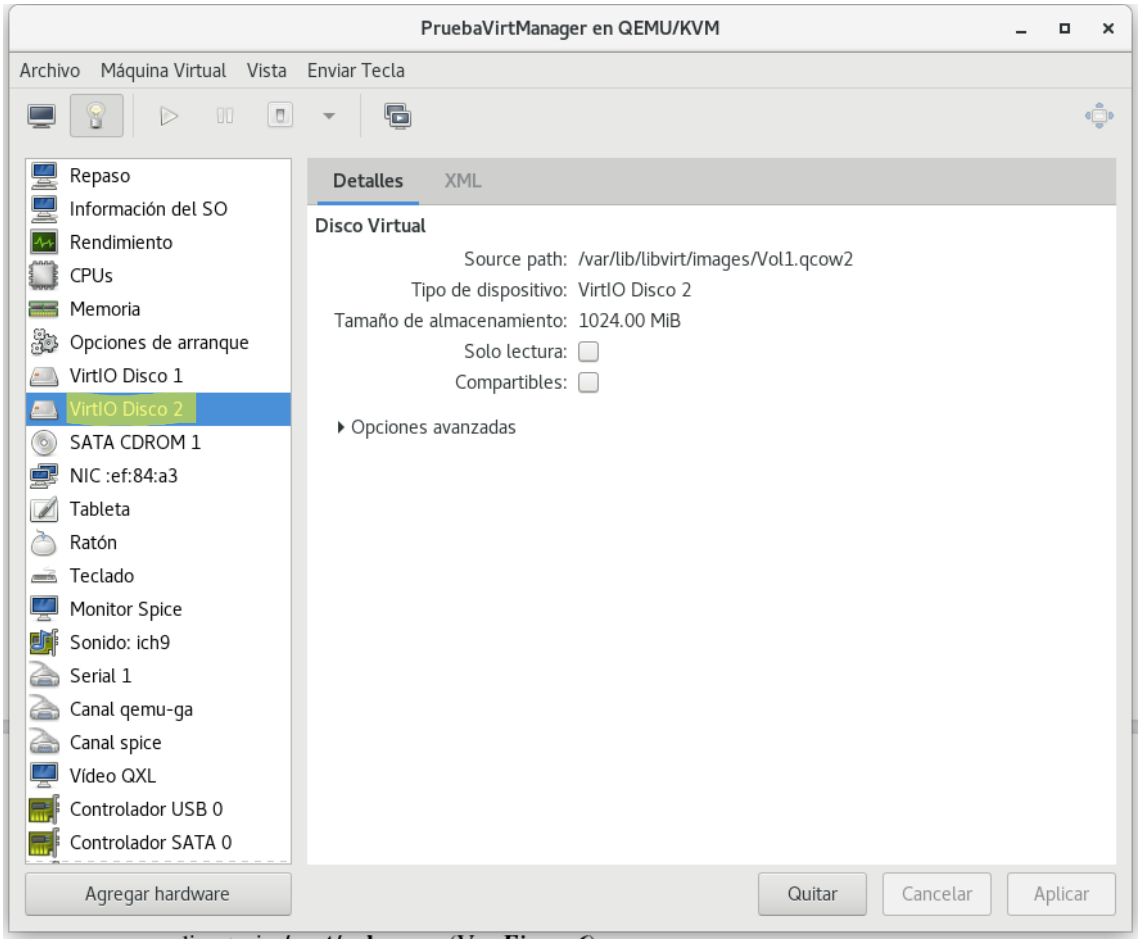

Ilustración 62: Detalles del hardware de una máquina virtual

Otra forma de comprobar que el volumen está asociado a la máquina virtual es iniciando la propia máquina virtual y ejecutar la orden "*lsblk*" para ver los discos. En la ***Ilustración 63*** se puede ver que el volumen creado es el etiquetado como "*vdb*".

```
[root@localhost ~]# lsblk -f
NAME        FSTYPE      LABEL UUID                                   MOUNTPOINT
sr0
vda
├─vda1      xfs               203fb989-e1ff-45ea-ba7b-fb37144fa7ac   /boot
└─vda2      LVM2_member       tnN16G-fsu6-NetN-hl8P-CDg1-6jo9-u9Jqqu
  ├─ol-root xfs               8ea26855-562c-439b-8613-92306e0c7f18   /
  └─ol-swap swap              d99a4727-a74f-498b-8b84-cfa5b29a25a9   [SWAP]
vdb
└─vdb1      ext4              43e244ff-6976-4bc4-ae3e-730fba091f7d   /mnt/pruebas
[root@localhost ~]# _
```

Ilustración 63: Discos en la máquina virtual PruebVirtManager

Por último, tras haber creado el volumen y haberlo asociado a una máquina virtual, se debe crear un fichero de prueba que demuestre que se pueden almacenar datos en el volumen desde la máquina virtual. Para ello, se creará un directorio para montar una partición en él (En este caso /mnt/pruebas) haciendo uso de la orden "*mkdir*".

Acto seguido, debemos particionar el volumen virtual que se ha asociado al host invitado (*vdb*) haciendo uso de alguna utilidad para particionar discos como "*fdisk*", "*cfdisk*", "*gdisk*", etc.

```
$ fdisk /dev/vdb
```

Tras haber creado la partición (*vdb1*), se procederá a crear en ella un sistema de archivos de tipo ext4:

```
$ mkfs.ext4 /dev/vdb1
```

Por último, se debe modificar el fichero *"/etc/fstab*", para que la partición *vdb1* del volumen *vdb* se monte en la carpeta creada anteriormente (/mnt/pruebas) de forma automática en el arranque de la máquina. Para ello, una vez modificado el fichero *fstab* se debe ejecutar la orden "*mount -a*". Tras el montaje, se debe crear algún fichero de prueba en la carpeta */mnt/pruebas* y comprobar que se están almacenando datos en el volumen creado y asociado a la máquina "*PruebaVirtManager*" correctamente, haciendo uso de la utilidad *"df*". Todos estos pasos se muestran en la ***Ilustración 64***:

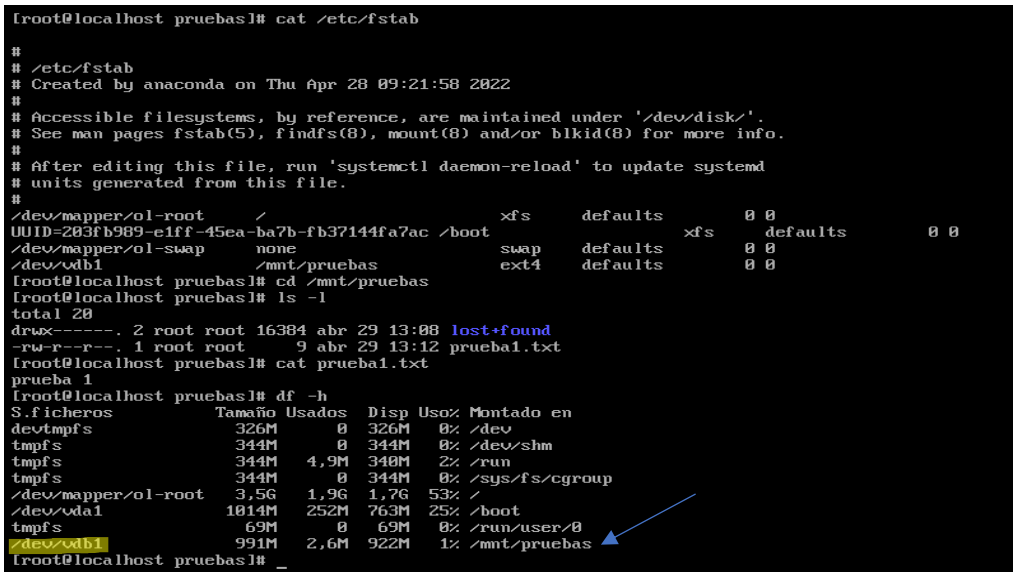

Ilustración 64: Comprobación de almacenamiento de datos en el volumen creado

Con Virsh:

Como se ha realizado anteriormente con la utilidad gráfica *virt-manager*, se creará un nuevo volumen de 1GB llamado *"Vol2"* en el contenedor por defecto "*default*" mediante la utilidad de línea de comandos "*virsh*" (Ver ***Ilustración 65***).

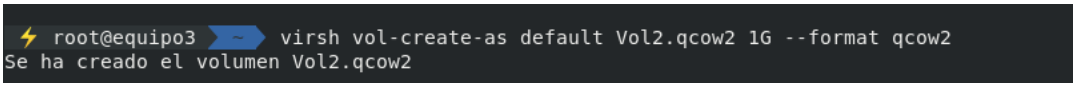

Ilustración 65: Creación de un volumen en el contenedor default con virsh

Acto seguido, se asociará el volumen "*Vol2*" recién creado a la máquina virtual "*Prueba_virt_install*" mediante *virsh* (Ver ***Ilustración 66***):

Ilustración 66: Asociación del volumen recién creado a la máquina virtual Prueba_virt_install mediante virsh

Además, podemos corroborar que el nuevo volumen ha sido creado con éxito mostrando la lista de volúmenes mediante *virsh* (Ver ***Ilustración 67***):

```
root@equipo3 ~ virsh vol-list default
 Nombre                          Ruta
------------------------------------------------------------------------------------------
 fedora_prueba.qcow2             /var/lib/libvirt/images/fedora_prueba.qcow2
 Prueba_clonacion.qcow2          /var/lib/libvirt/images/Prueba_clonacion.qcow2
 Prueba_copiando_ficheros.qcow2  /var/lib/libvirt/images/Prueba_copiando_ficheros.qcow2
 Prueba_virt_clone.qcow2         /var/lib/libvirt/images/Prueba_virt_clone.qcow2
 Prueba_virt_install.qcow2       /var/lib/libvirt/images/Prueba_virt_install.qcow2
 PruebaVirtManager.qcow2         /var/lib/libvirt/images/PruebaVirtManager.qcow2
 Vol1.qcow2                      /var/lib/libvirt/images/Vol1.qcow2
 Vol2.qcow2                      /var/lib/libvirt/images/Vol2.qcow2
```

Ilustración 67: Comprobación de la creación del volumen Vol2 en el listado de volúmenes del contenedor default

Tras haber realizado los pasos anteriores, se tendrá un nuevo volumen de 1GB asociado a la máquina *Prueba_virt_install* mediante la utilidad de línea de órdenes *virsh*. Por último, se debe comprobar que se pueden almacenar datos en el volumen desde la máquina virtual, realizando los mismos pasos que se realizaron en el apartado anterior de "*Creación y asociación de un volumen a una máquina virtual*" mediante *virt-manager.*

Anexo VIII: Pasos para la creación de una nueva partición lógica en el host anfitrión y asociación a una MV

Antes que nada, el disco físico "*sda*" del sistema anfitrión cuenta con dos particiones: *sda1* (partición primaria de 1K) y *sda2* (partición extendida de 500M), como se muestra en la ***Ilustración 68***:

```
root@equipo3 ~ lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0   1,8T  0 disk
├─sda1   8:1    0   500M  0 part
└─sda2   8:2    0     1K  0 part
sdb      8:16   0   1,8T  0 disk
└─sdb1   8:17   0   1,8T  0 part
sdc      8:32   0 223,6G  0 disk
├─sdc1   8:33   0  45,4G  0 part
├─sdc2   8:34   0    1G  0 part
├─sdc3   8:35   0  98,5G  0 part /
├─sdc4   8:36   0    1K  0 part
├─sdc5   8:37   0   10G  0 part /home
└─sdc6   8:38   0  1023M  0 part /boot
```

Ilustración 68: Particionado del sistema anfitrión

A continuación, se procederá con la creación de una nueva partición lógica de tipo Linux (*sda5*) sobre la partición extendida del disco físico ***sda*** del sistema anfitrión (*sda2*) de 1GB de capacidad. Para la creación de esta nueva partición lógica, se hará uso de la utilidad "*fdisk*" (Ver ***Ilustración 69***):

```
root@equipo3 ~ fdisk /dev/sda

Bienvenido a fdisk (util-linux 2.32.1).
Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.
Tenga cuidado antes de utilizar la orden de escritura.


Orden (m para obtener ayuda): p
Disco /dev/sda: 1,8 TiB, 2000398934016 bytes, 3907029168 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 4096 bytes
Tamaño de E/S (mínimo/óptimo): 4096 bytes / 4096 bytes
Tipo de etiqueta de disco: dos
Identificador del disco: 0x802bccf2

Disposit.  Inicio Comienzo    Final Sectores Tamaño Id Tipo
/dev/sda1  *          2048  1026047  1024000   500M  7 HPFS/NTFS/exFAT
/dev/sda2          1026048 42969087 41943040    20G  5 Extendida

Orden (m para obtener ayuda): n
Tipo de partición
   p   primaria (1 primaria(s), 1 extendida(s), 2 libre(s))
   l   lógica (la numeración empieza por 5)
Seleccionar (valor predeterminado p): l

Se añade la partición lógica 5
Primer sector (1028096-42969087, valor predeterminado 1028096):
Último sector, +sectores o +tamaño{K,M,G,T,P} (1028096-42969087, valor predeterminado 42969087): +1G

Crea una nueva partición 5 de tipo 'Linux' y de tamaño 1 GiB.

Orden (m para obtener ayuda): p
Disco /dev/sda: 1,8 TiB, 2000398934016 bytes, 3907029168 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 4096 bytes
Tamaño de E/S (mínimo/óptimo): 4096 bytes / 4096 bytes
Tipo de etiqueta de disco: dos
Identificador del disco: 0x802bccf2

Disposit.  Inicio Comienzo    Final Sectores Tamaño Id Tipo
/dev/sda1  *          2048  1026047  1024000   500M  7 HPFS/NTFS/exFAT
/dev/sda2          1026048 42969087 41943040    20G  5 Extendida
/dev/sda5          1028096  3125247  2097152     1G 83 Linux

Orden (m para obtener ayuda): w
Se ha modificado la tabla de particiones.
Llamando a ioctl() para volver a leer la tabla de particiones.
Se están sincronizando los discos.
```

Ilustración 69: Creación de una partición lógica de 1GB en el sistema anfitrión

Tras la creación de la nueva partición lógica de 1GB, habrá que reiniciar el sistema para que la nueva partición se cargue adecuadamente (*reboot*). Una vez arrancado de nuevo el sistema, se puede comprobar la existencia de la nueva partición lógica recién creada con las utilidades "*lsblk*" y "*fdisk*".

Acto seguido, se deberá asociar dicha partición lógica a una máquina virtual de KVM. Para ello, se debe crear un fichero XML que indique las especificaciones del dispositivo físico de bloque a asociar (sda5, en este caso).

El fichero XML, puede tener la siguiente apariencia (Ver ***Ilustración 70***):

```
root@equipo3 ~ cat disco_fisico.xml
<disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none'/>
      <source dev='/dev/sda5'/>
      <target dev='vdc'/>
</disk>
root@equipo3 ~
```

Ilustración 70: Fichero XML con las especificaciones de la partición lógica sda5

Una vez creado el fichero XML, se debe hacer uso de la utilidad *virsh* para asociar la partición lógica a una máquina virtual (En este caso *Prueba_virt_install*) de la siguiente forma (Ver ***Ilustración 71***):

```
root@equipo3 ~ virsh attach-device --config Prueba_virt_install disco_fisico.xml
El dispositivo fue asociado exitosamente
```

Ilustración 71: Asociación de una partición lógica a una máquina virtual mediante virsh

Una vez asociada la partición lógica del host a la máquina virtual "*Prueba_virt_install"*, se puede comprobar que el proceso se ha efectuado correctamente observando los detalles del hardware de la propia máquina virtual y observando que aparece un tercer disco que hace referencia a la partición lógica ***sda5*** del sistema host anfitrión (Ver ***Ilustración 72***):

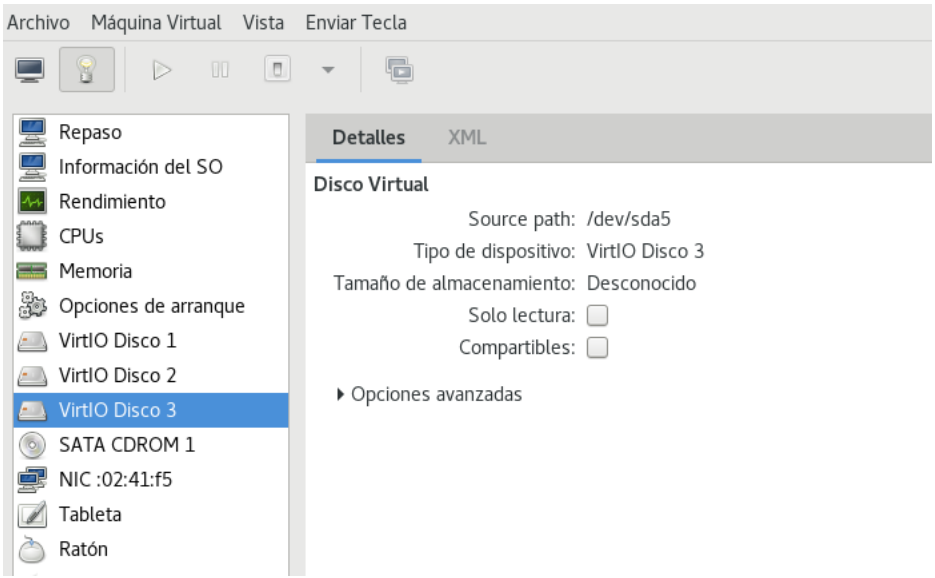

Ilustración 72: Comprobación del asociamiento de la partición lógica sda5 a la máquina virtual

Por último, se debe comprobar que se pueden almacenar datos en el volumen que hace referencia a la partición lógica del host anfitrión (*vdc*) desde la máquina virtual. Para ello, se pueden seguir los mismos pasos realizados en anexos anteriores.

Anexo IX: Pasos para la creación de un contenedor FS sobre una partición lógica del sistema anfitrión y un volumen en su interior

En primer lugar, se procederá con la creación de una nueva partición lógica de tipo Linux (*sda6*) sobre la partición extendida del disco físico ***sda*** del sistema anfitrión (*sda2*) con una capacidad de 2GB. Para la creación de esta nueva partición lógica, se hará uso del comando "*fdisk /dev/sda*" como en el anexo anterior (Ver Anexo VIII).

Una vez creada la partición lógica de 2GB, se reiniciará el sistema (*reboot*) y se comprobará con la utilidad "*lsblk*" que la partición fue creada con éxito. Tras esta comprobación, se le dará formato a esta nueva partición lógica (sda6), creando en ella un sistema de archivos *ext4* de la siguiente manera:

```
$ mkfs.ext4 /dev/sda6
```

Con la nueva partición lógica de 2GB (sda6) creada y formateada, se procederá con la creación del contenedor FS haciendo uso primero, de *virt-manager* y luego de la utilidad *virsh*.

Con Virt-manager:

Navegar a "*Editar-Detalles de la conexión-Almacenamiento*" y hacer clic en el icono que está en la parte inferior izquierda. Al crear un nuevo contenedor/silo, se debe especificar el nombre del contenedor (Contenedor_Particion), el tipo de contenedor a crear, su formato, la dirección fuente (en este caso la partición sda6) y la ruta especificada para el grupo de almacenamiento (*Target Path*). En la ***Ilustración 73*** se puede ver los detalles especificados para la creación de este nuevo contendor de Dispositivo de Bloque Preformateado sobre la partición lógica ***sda6*** del sistema anfitrión.

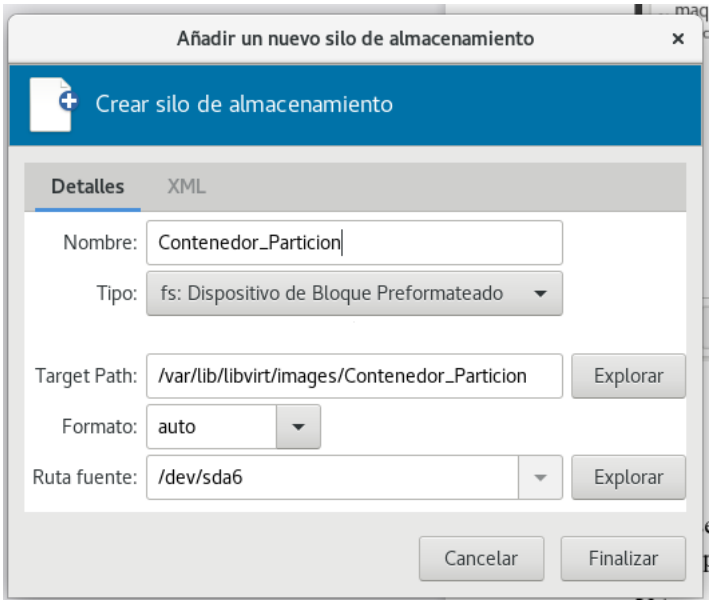

Ilustración 73: Creación del contenedor FS

Una vez creado el nuevo contenedor de tipo FS llamado "Contenedor_Particion" sobre la partición lógica ***sda6*** del sistema host anfitrión, podemos corroborar su existencia con la interfaz-gráfica *virt-manager* (Ver ***Ilustración 74***):

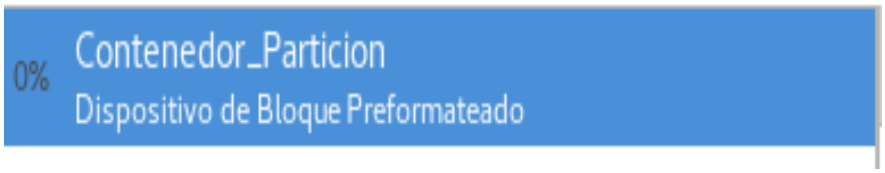

Ilustración 74: Verificación de la creación del contenedor FS sobre la partición lógica de 2GB del host anfitrión

Y también podemos comprobar haciendo uso de la orden "*lsblk*" que el contenedor está montado en la partición lógica ***sda6*** del host.

Con Virsh:

Acto seguido, se realizará el mismo trabajo que se ha realizado con *virt-manager* en el apartado anterior, pero haciendo uso esta vez de la utilidad de línea de comandos "*virsh*".

Lo primero que se debe de hacer será crear una nueva partición lógica de 2GB de tipo Linux sobre la partición extendida del disco físico ***sda*** del host anfitrión (*sda2*), haciendo uso de la utilidad "*fdisk*" y tal y como se ha visto anteriormente.

Una vez creada la partición lógica (sda7), se reiniciará el sistema (*reboot*) y se comprobará con la utilidad "*lsblk*" que la partición fue creada con éxito. Tras esta comprobación, se le dará formato a esta nueva partición lógica (sda7), creando en ella un sistema de archivos *ext4* de la siguiente manera:

```
$ mkfs.ext4 /dev/sda7
```

Una vez creada y formateada la nueva partición lógica (*sda7*), se procederá con la creación mediante *virsh* de un nuevo contenedor de tipo FS. Para ello, se deberá crear un nuevo fichero XML con una apariencia similar al de la ***Ilustración 75***, en dónde se especifican los parámetros para la creación de este nuevo contenedor de Dispositivo de Bloque Preformateado, tales como: su nombre (Contenedor_Virsh), ruta fuente, su tipo, formato, etc.

```
<pool type="fs">
  <name>Contenedor_Virsh</name>
  <uuid>f0f8701a-4227-471c-8c50-a2006149a919</uuid>
  <capacity unit="bytes">2046640128</capacity>
  <allocation unit="bytes">6291456</allocation>
  <available unit="bytes">2040348672</available>
  <source>
    <device path="/dev/sda7"/>
    <format type="auto"/>
  </source>
  <target>
    <path>/var/lib/libvirt/images/Contenedor_Virsh</path>
    <permissions>
      <mode>0755</mode>
      <owner>0</owner>
      <group>0</group>
      <label>system_u:object_r:unlabeled_t:s0</label>
    </permissions>
  </target>
</pool>
```

Ilustración 75: Fichero XML con las especificaciones del contenedor FS a crear con virsh

En primer lugar, se usará el fichero XML creado anteriormente para definir el nuevo contenedor FS. Para ello se ejecutará (Ver ***Ilustración 76***):

```
root@equipo3 ~ virsh pool-define contenedor_fs.xml
El grupo Contenedor_Virsh ha sido definido desde contenedor_fs.xml
```

Ilustración 76: Definición del contendor FS con virsh

En segundo lugar, se compilará el contenedor definido anteriormente de la siguiente manera (Ver ***Ilustración 77***):

```
root@equipo3 ~ virsh pool-build Contenedor_Virsh
El pool Contenedor_Virsh ha sido compilado
```

Ilustración 77: Compilación del contenedor "Contenedor_Virsh"

Por último, se marcará el nuevo contenedor recién creado para que se inicie automáticamente en el arranque del sistema (Ver ***Ilustración 78***):

```
root@equipo3 ~ virsh pool-autostart Contenedor_Virsh
El grupo Contenedor_Virsh ha sido marcado como iniciable automáticamente
```

Ilustración 78: Contenedor FS marcado como iniciable automáticamente con virsh

Creación de un volumen en el contenedor FS y asociación a una MV:

A continuación, se procederá a crear un nuevo volumen de 1GB en el contenedor de tipo FS creado anteriormente. Para la creación de este nuevo volumen, se hará uso de la utilidad "*virsh*" (Ver ***Ilustración 79***):

```
root@equipo3 ~ virsh vol-create-as Contenedor_Particion Vol3.qcow2 1G --format qcow2
Se ha creado el volumen Vol3.qcow2
```

Ilustración 79: Creación de un volumen de 1GB en el Contenedor_Particion

Una vez creado el volumen en el contenedor FS, se asociará con una máquina virtual, en este caso con la máquina "Prueba_clonacion", haciendo uso de la orden "*virsh attach-disk*" (Ver ***Ilustración 80***):

```
root@equipo3 ~ virsh attach-disk --config Prueba_clonacion --source /var/lib/libvirt/images/Vol3.qcow2 --target vdb --cache none --subdriver qcow2
El disco ha sido asociado exitosamente
```

Ilustración 80: Asociación del volumen de 1GB del contenedor FS a una MV

Podemos corroborar que el volumen ha sido creado exitosamente en el *Contenedor_Particion* con el siguiente comando (Ver ***Ilustración 81***):

```
root@equipo3 ~ virsh vol-list Contenedor_Particion
 Nombre       Ruta
------------------------------------------------------------------------------
 lost+found   /var/lib/libvirt/images/Contenedor_Particion/lost+found
 Vol3.qcow2   /var/lib/libvirt/images/Contenedor_Particion/Vol3.qcow2

root@equipo3 ~
```

Ilustración 81: Listado de volúmenes del Contenedor_Particion

Por último, se debe comprobar que se pueden almacenar datos en el volumen desde la máquina virtual. Para ello, se pueden seguir los mismos pasos realizados en anexos anteriores.

Anexo X: Configuración del servidor NFS simulado que exporta recursos al host anfitrión

Como se ha mencionado en el capítulo XII, se ha configurado un servicio NFS simulado por una máquina virtual KVM que proporciona diferentes recursos al host anfitrión.

Para la configuración de este servidor NFS, se ha tenido que instalar el propio servicio NFS en la máquina virtual. Para ello, se han instalado todas las utilidades de este servicio en la MV de la siguiente manera:

```
$ dnf -y install nfs-utils
```

Una vez instaladas las utilidades de NFS, se deben iniciar y dejar habilitadas en el arranque de este servidor NFS simulado, el demonio *rpc-bind* y el propio servidor NFS:

```
$ systemctl enable --now nfs-server
```

```
$ systemctl enable --now rpcbind
```

Nota: El flag ***"--now"*** inicia el servicio además de habilitarse en el arranque (es similar a ejecutar "systemctl start nombre_servicio").

Por defecto, SELinux no permite que NFS comparta directorios, por lo que será necesario activar las banderas booleanas (*flags*) adecuadas para permitirlo. Además, se debe tomar en cuenta que será necesario activar las banderas para permitir lecturas y escrituras en los directorios exportados. Para ello, se deberá activar el siguiente booleano de SELinux en el servidor NFS simulado:

```
$ setsebool -P nfs_export_all_rw 1
```

Además, se deben añadir los servicios **nfs**, **rpc-bind** y **mountd** al cortafuegos *firewalld* de manera que se apliquen al conjunto permanente del cortafuegos (--permanent) para el correcto funcionamiento del servicio NFS:

```
$ firewall-cmd --permanent --add-service nfs
```

```
$ firewall-cmd --permanent --add-service rpc-bind
```

```
$ firewall-cmd --permanent --add-service mountd
```

Hacemos los cambios persistentes en el conjunto en ejecución:

```
$ firewall-cmd --reload
```

Una vez llegados a este punto, se modificó el archivo "***/etc/exports***" de manera que se permita exportar los directorios ***/volumenes*** e ***/ISOS/OL/8.5*** del servidor NFS al host anfitrión. Estos directorios contienen volúmenes e imágenes ISOS de sistemas operativos, respectivamente. En la ***Ilustración 79*** se puede observar el contenido del fichero ***"/etc/exports"*** del servidor NFS:

```
/volumenes *(rw, sync, no_root_squash, no_all_squash)
/ISOS/OL/8.5 *(rw, sync, no_root_squash, no_all_squash)
```

Ilustración 82: Contenido del fichero /etc/exports del servidor NFS simulado

Nota: El comodín **"*"** significa que los directorios */volumenes* e */ISOS/OL/8.5* se pueden exportar a cualquier nombre de dominio, con permisos de escritura (rw), lectura, de sincronización (sync), etc.

Una vez modificado el fichero "***/etc/exports"***, se debe exportar su contenido ejecutando la orden:

```
$ exportfs -arv
```

Tras exportar su contenido, se deben reiniciar los servicios ***rpcbind*** y ***nfs-server***, para que se apliquen los cambios:

```
$ systemctl restart nfs-server rpcbind
```

Se pueden observar los directorios exportados por el servidor NFS simulado, haciendo uso de la orden "*showmount*" como se muestra en la ***Ilustración 83***:

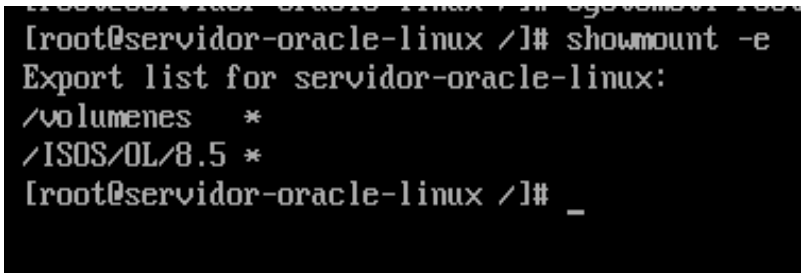

Ilustración 83: Directorios exportados por el servicio NFS simulado

Por último, se debe activar el booleano "***virt_use_nfs***" en el sistema anfitrión para permitir que el software de virtualización tenga acceso a sistemas de archivos NFS:

```
$ setsebool -P virt_use_nfs on
```

Y modificar el fichero "***/etc/hosts***" del sistema host anfitrión para poder conectarse al servidor NFS simulado por su nombre de dominio completamente cualificado (*FQDN*), además de por su dirección IP (192.168.122.237).

Ilustración 84: Entrada en el fichero hosts del sistema anfitrión para permitir la conexión por el nombre de dominio del servidor NFS simulado

Nota: La dirección IP 192.168.122.237 es la dirección de la máquina virtual KVM que simula el servicio NFS.

Anexo XI: Creación del contenedor NFS de imágenes ISO con *virt-manager*

Para crear el nuevo contenedor NFS, se navegará a ***"Editar-Detalles de la conexión-Almacenamiento"*** y se hará clic en el icono que está en la parte inferior izquierda. Al crear un contenedor NFS se tendrá que especificar su nombre (En este caso, *Contenedor_NFS*), su tipo (*netfs*), su formato, la ruta fuente (que hará referencia al directorio exportado por el servicio NFS que ofrece imágenes ISO *"/ISOS/ OL/8.5*"), la ruta de destino en el sistema y el nombre del equipo que proporcionará el servicio NFS (En este caso, *servidor-oracle-linux*). Todas estas especificaciones se muestran en la ***Ilustración 85***:

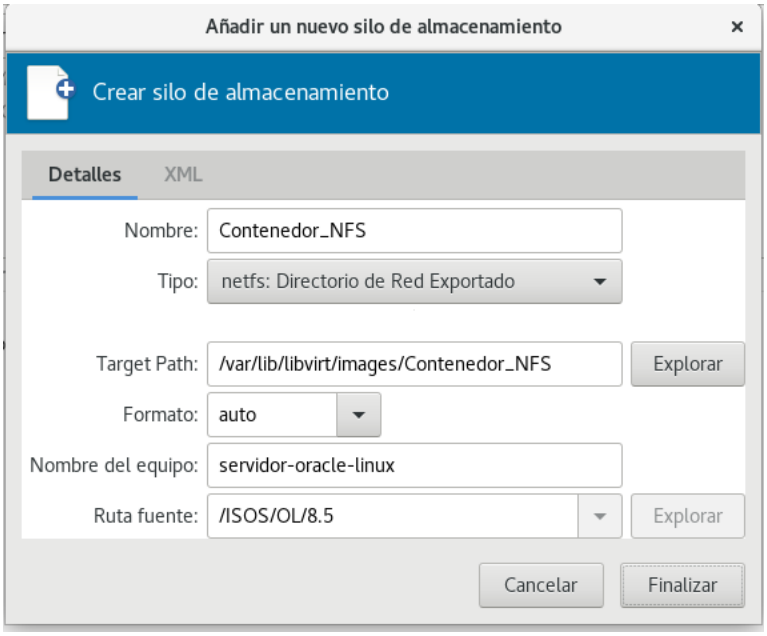

Ilustración 85: Creación de un contenedor NFS de imágenes ISO

Tras hacer clic en el botón “Finalizar”, se creará el nuevo contenedor NFS que proporciona una imagen ISO de Oracle Linux al sistema host anfitrión (Ver ***Ilustración 86***):

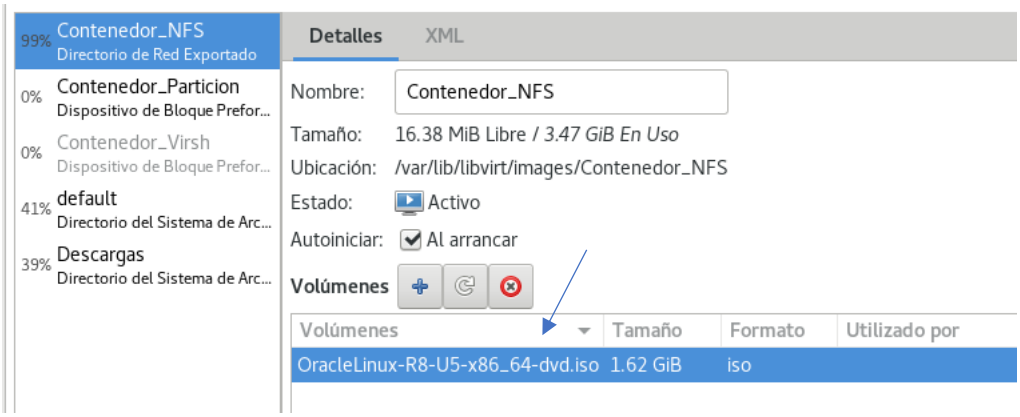

Ilustración 86: Comprobación de la creación del contenedor NFS que proporciona imágenes ISO al host anfitrión

Anexo XII: Creación del contenedor NFS de volúmenes de máquinas virtuales

Para la creación de este segundo contenedor NFS para volúmenes de máquinas virtuales se utilizará la utilidad de línea de órdenes "*virsh*", por lo que primero que nada se debe crear un fichero XML con las especificaciones del contenedor NFS a crear (Ver ***Ilustración 87***). En este fichero debe estar especificado la ruta fuente (en este caso el directorio */volumenes* proporcionado por el servicio NFS simulado), el tipo de contenedor a crear (*netfs*), el nombre del servidor NFS (En este caso "*servidor-oracle-linux"*), etc.

```
<pool type="netfs">
  <name>Contenedor_NFS_Volumenes</name>
  <uuid>dfebaa84-014a-4601-ae69-ab397220d359</uuid>
  <source>
    <host name="servidor-oracle-linux"/>
    <dir path="/volumenes"/>
    <format type="auto"/>
  </source>
  <target>
    <path>/var/lib/libvirt/images/Contenedor_NFS_Volumenes</path>
    <permissions>
      <mode>0755</mode>
      <owner>0</owner>
      <group>0</group>
      <label>system_u:object_r:nfs_t:s0</label>
    </permissions>
  </target>
</pool>
```

Ilustración 87: Fichero XML con las especificaciones del contenedor NFS para volúmenes de máquinas virtuales

Una vez creado el fichero XML con todas las especificaciones necesarias, se procederá a utilizar la utilidad "*virsh*" para definir el nuevo contenedor NFS a partir del fichero XML, compilarlo y marcarlo como iniciable automáticamente en el arranque del sistema (Ver ***Ilustración 88***).

```
root@equipo3 ~ virsh pool-define contenedor_NFS.xml
El grupo Contenedor_NFS_Volumenes ha sido definido desde contenedor_NFS.xml

root@equipo3 ~ virsh pool-build Contenedor_NFS_Volumenes
El pool Contenedor_NFS_Volumenes ha sido compilado

root@equipo3 ~ virsh pool-autostart Contenedor_NFS_Volumenes
El grupo Contenedor_NFS_Volumenes ha sido marcado como iniciable automáticamente
```

Ilustración 88: Creación del contenedor NFS para máquinas virtuales mediante virsh

Tras la creación del contenedor NFS, se procederá a crear un nuevo volumen de 1GB en él. La creación de este nuevo volumen (*Vol4*) se llevará a cabo con *virt-manager (*Ver ***Ilustración 89***)*:*

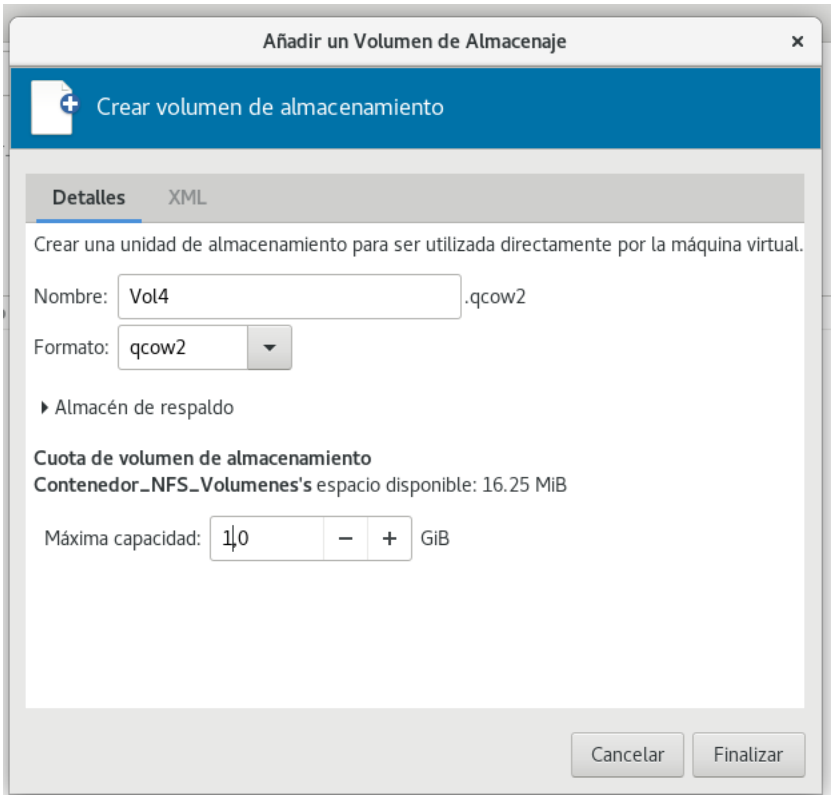

Ilustración 89: Creación de un volumen de 1GB en el contenedor NFS para volúmenes de máquinas virtuales

Podemos corroborar la creación del volumen recién creado en el contenedor NFS haciendo uso de *virt-manager* (Ver ***Ilustración 90***):

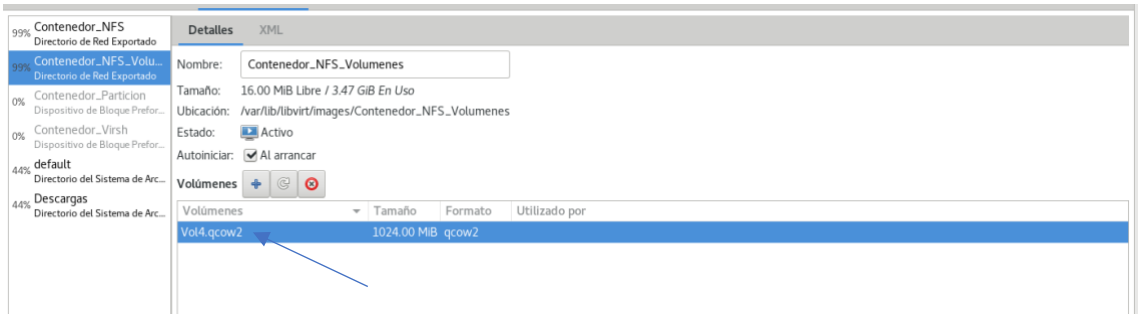

Ilustración 90: Comprobación de la creación del volumen Vol4 en el contenedor NFS para volúmenes de MV

El contenedor NFS permite que las máquinas virtuales almacenen volúmenes en otro servidor externo (Servidor NFS) y así proporcionar más espacio de almacenamiento a las máquinas virtuales. Si desde otra máquina se creara un contenedor NFS con los datos del servidor NFS "***servidor-oracle-linux***" y se utilizará como ruta fuente el directorio ***"/volumenes***" del servicio NFS simulado, se tendría acceso al volumen denominado "***Vol4***" creado anteriormente y viceversa.

Anexo XIII: Pasos para la creación del bridge en la interfaz física del host anfitrión y conexión de una MV al bridge

En primer lugar, se creará en el host anfitrión el fichero para la creación del bridge (ifcfg-br0). Este fichero estará bajo el directorio *"/etc/sysconfig/network-scripts*" y su contenido será el siguiente (Ver ***Ilustración 91***):

ifcfg-br0

```
DEVICE=br0
BOOTPROTO=dhcp
ONBOOT=yes
TYPE=Bridge
```

Ilustración 91: Contenido del fichero de creación del bridge

Tras la creación del bridge, se debe modificar el fichero de la interfaz física del host anfitrión indicándole que forma parte del bridge (Ver ***Ilustración 92***):

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=dhcp
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
NAME=enp0s25
UUID=64618dab-e0fc-4a9b-ac8c-badbbe0807a9
DEVICE=enp0s25
ONBOOT=yes
NM_CONTROLLED=no
BRIDGE=br0
```

Ilustración 92: Fichero de la interfaz física del host anfitrión

Acto seguido, se debe reiniciar la red en el host anfitrión para que se apliquen los cambios:

```
$ systemctl restart network
```

Podemos observar hacienda uso del comando "*ifconfig*", como la dirección IP de la interfaz física del host anfitrión está asignada a la interfaz "br0" (Ver ***Ilustración 93***):

```
root@equipo3 .../sysconfig/network-scripts ifconfig
br0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.13.15.18  netmask 255.255.255.0  broadcast 10.13.15.255
        inet6 fe80::222:4dff:fe6a:26b0  prefixlen 64  scopeid 0x20<link>
        ether 00:22:4d:6a:26:b0  txqueuelen 1000  (Ethernet)
        RX packets 372  bytes 73222 (71.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 225  bytes 24581 (24.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Ilustración 93: Asignación de la IP de la interfaz física del host a la interfaz br0

A continuación, se creará una nueva máquina virtual en KVM y se conectará al bridge haciendo uso de *virt-manager* (Ver ***Ilustración 94***):

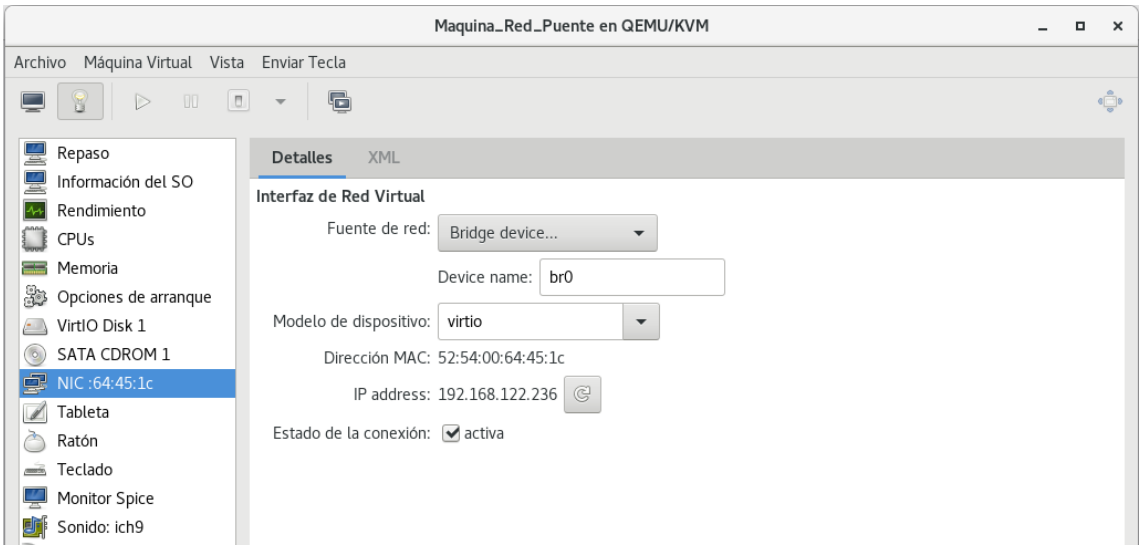

Ilustración 94: Conexión de una MV al bridge

Al iniciar la máquina virtual, se debe registrar su dirección MAC en el servidor DHCP y configurar el fichero de la interfaz de la máquina virtual mediante DHCP (Ver ***Ilustración 95***):

```
[root@localhost network-scripts]# cat ifcfg-enp1s0
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=dhcp
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
NAME=enp1s0
UUID=6c393e4f-6c52-4084-8b91-fd3b68816034
DEVICE=enp1s0
ONBOOT=yes
[root@localhost network-scripts]# _
```

Ilustración 95: Fichero de configuración de red de la interfaz de la máquina virtual conectada al bridge

Por último, podemos comprobar haciendo uso de la orden "*ip a*" que la dirección IP de la interfaz de red de la máquina virtual está en la misma red que la interfaz física del host anfitrión (10.13.15.0/24):

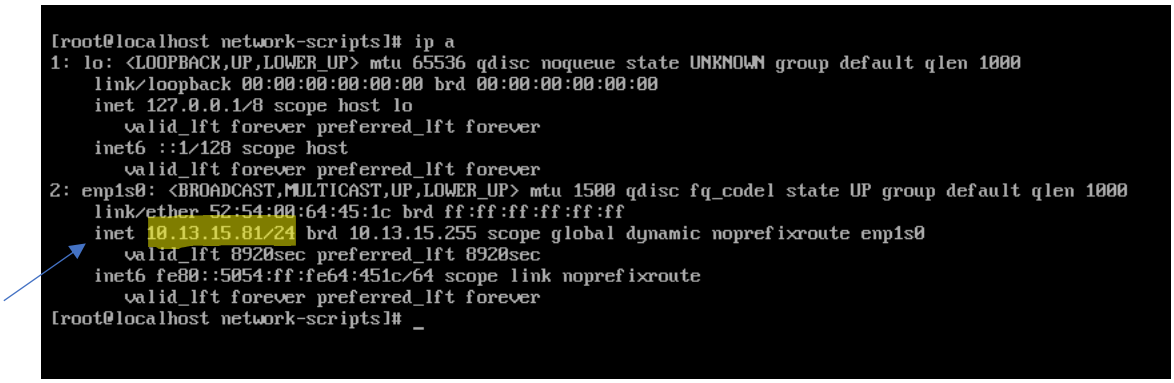

Ilustración 96: Interfaz de red de la MV en la misma red que la interfaz física del sistema anfitrión

Anexo XIV: Creación de la Red aislada de Almacenamiento

Para crear estar red virtual, se accederá a KVM mediante *virt-manager* y se navegará a "***Editar-Detalles de la conexión-Redes virtuales***" y se hará clic en el icono "**+**" de color azul que aparece en la parte inferior izquierda. Al crear una red en KVM se tendrá que indicar: nombre de la red (en este caso "Almacenamiento"), el modo en el que opera (en este caso en modo aislada), direcciones de red (en este caso "10.22.122.0/24"), indicar si se usa o no servicio DHCP (en este caso "no"), etc. (Ver ***Ilustración 97***).

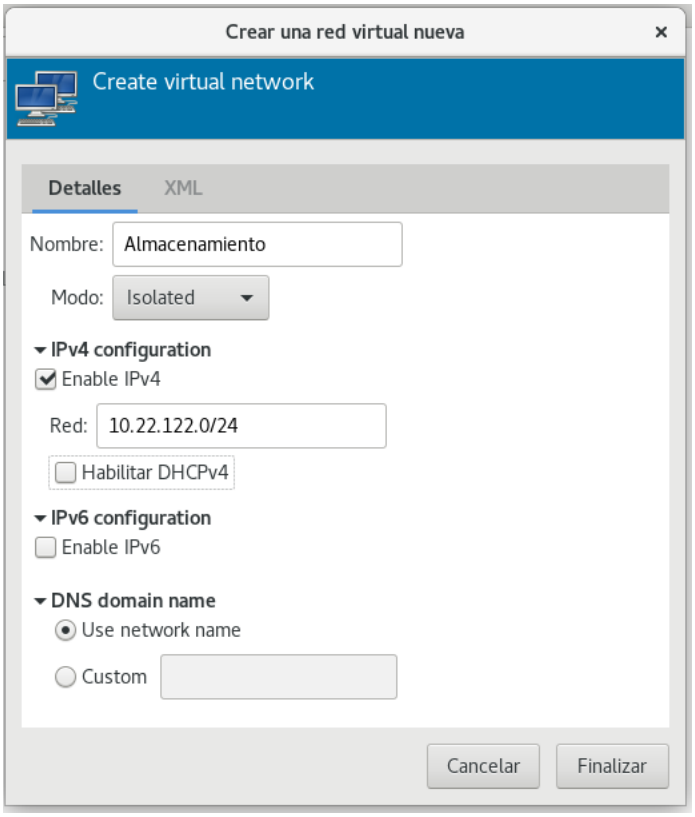

Ilustración 97: Creación de la Red aislada de Almacenamiento del clúster

Anexo XV: Creación de la Red aislada de Control

Para crear estar red virtual, se accederá a KVM mediante *virt-manager* y se navegará a "***Editar-Detalles de la conexión-Redes virtuales***" y se hará clic en el icono "**+**" de color azul que aparece en la parte inferior izquierda. Al crear una red en KVM se tendrá que indicar: nombre de la red (en este caso "Control"), el modo en el que opera (en este caso en modo aislada), direcciones de red (En este caso 10.22.132.0/24), indicar si se usa o no servicio DHCP (En este caso "no"), etc. (Ver ***Ilustración 98***).

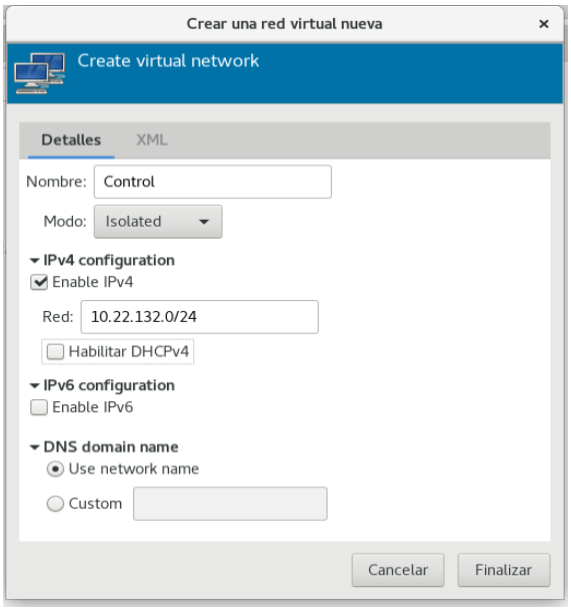

Ilustración 98: Creación de la Red aislada de Almacenamiento del clúster

Anexo XVI: Añadir un nuevo disco de almacenamiento a una MV

Para añadir un nuevo disco de almacenamiento a una máquina virtual mediante *virt-manager*, tan solo se tendrá que navegar a "***Editar-Detalles de la máquina virtual-Panel de Hardware-Agregar Hardware-Almacenamiento***" (Ver ***Ilustración 99***):

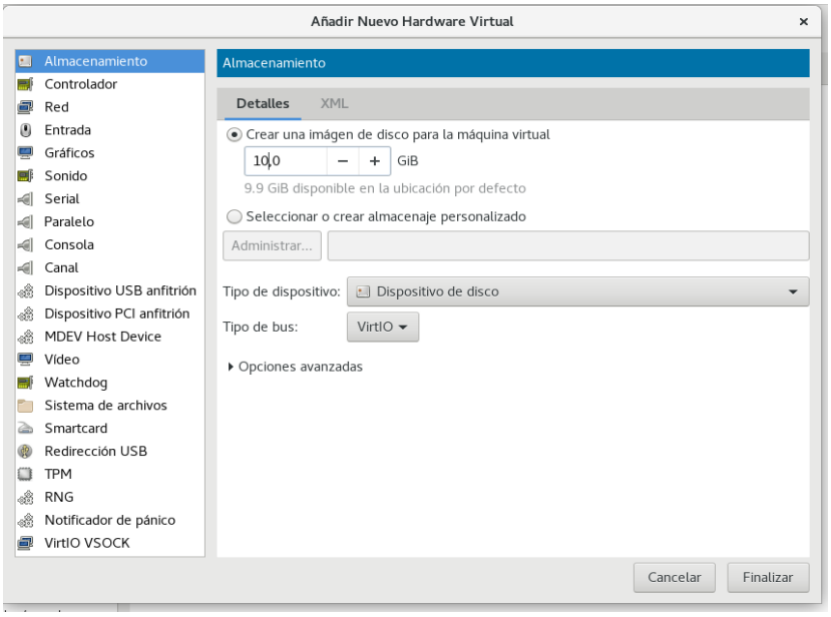

Ilustración 99: Añadir disco de almacenamiento a una MV

Anexo XVII: Interfaces de red del Nodo1

Las interfaces de red del Nodo1 tras su configuración son:

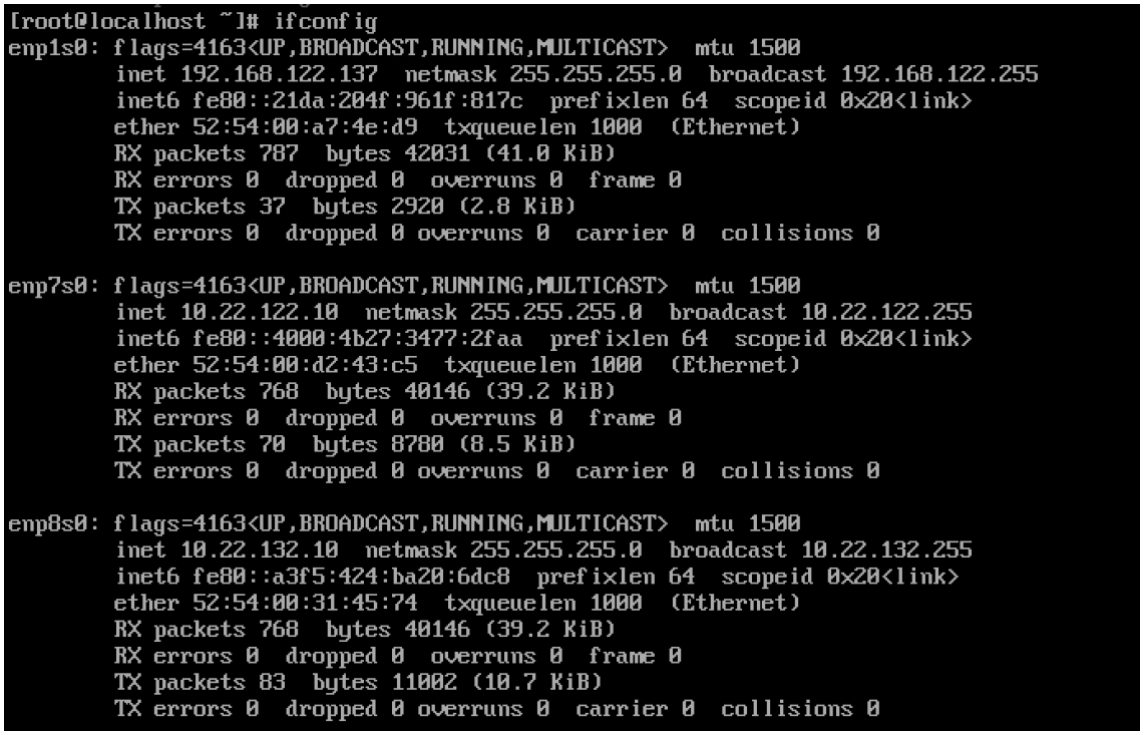

Ilustración 100: Interfaces de red del Nodo1

Anexo XVIII: Interfaces de red del Nodo2

Las interfaces de red del Nodo2 tras su configuración son:

```
[root@localhost ~]# ifconfig
enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.122.124  netmask 255.255.255.0  broadcast 192.168.122.255
        inet6 fe80::490c:69b7:ebba:efa6  prefixlen 64  scopeid 0x20<link>
        ether 52:54:00:d3:9e:8e  txqueuelen 1000  (Ethernet)
        RX packets 204  bytes 15299 (14.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 39  bytes 3666 (3.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp7s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.22.122.11  netmask 255.255.255.0  broadcast 10.22.122.255
        inet6 fe80::6ddd:ce3d:173d:9b1a  prefixlen 64  scopeid 0x20<link>
        ether 52:54:00:c5:c3:f4  txqueuelen 1000  (Ethernet)
        RX packets 173  bytes 12055 (11.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 66  bytes 8532 (8.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp8s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.22.132.11  netmask 255.255.255.0  broadcast 10.22.132.255
        inet6 fe80::6013:5eb1:d09c:e71  prefixlen 64  scopeid 0x20<link>
        ether 52:54:00:8b:53:a7  txqueuelen 1000  (Ethernet)
        RX packets 173  bytes 12055 (11.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 77  bytes 10630 (10.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@localhost ~]# _
```

Ilustración 101: Interfaces de red del Nodo2

Anexo XIX: Interfaces de red del Nodo3

Las interfaces de red del Nodo3 tras su configuración son:

```
[root@localhost ~]# ifconfig
enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.122.199  netmask 255.255.255.0  broadcast 192.168.122.255
        inet6 fe80::eb6a:2e7d:1eea:f3c1  prefixlen 64  scopeid 0x20<link>
        ether 52:54:00:7f:d1:9d  txqueuelen 1000  (Ethernet)
        RX packets 167  bytes 12724 (12.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 32  bytes 3129 (3.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp7s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.22.122.12  netmask 255.255.255.0  broadcast 10.22.122.255
        inet6 fe80::ddcc:fa1e:a1a6:79fd  prefixlen 64  scopeid 0x20<link>
        ether 52:54:00:b6:f0:ce  txqueuelen 1000  (Ethernet)
        RX packets 148  bytes 10256 (10.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 50  bytes 6144 (6.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp8s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.22.132.12  netmask 255.255.255.0  broadcast 10.22.132.255
        inet6 fe80::d5cb:6820:6a0f:a002  prefixlen 64  scopeid 0x20<link>
        ether 52:54:00:1c:91:99  txqueuelen 1000  (Ethernet)
        RX packets 148  bytes 10256 (10.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 54  bytes 7456 (7.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@localhost ~]#
```

Ilustración 102: Interfaces de red del Nodo3

Anexo XX: Interfaces de red del Nodo4

Las interfaces de red del Nodo4 tras su configuración son:

Ilustración 103: Interfaces de red del Nodo4

Anexo XXI: Interfaces de red del Balanceador-De-Carga

Las interfaces de red del Balanceador-De-Carga tras su configuración son:

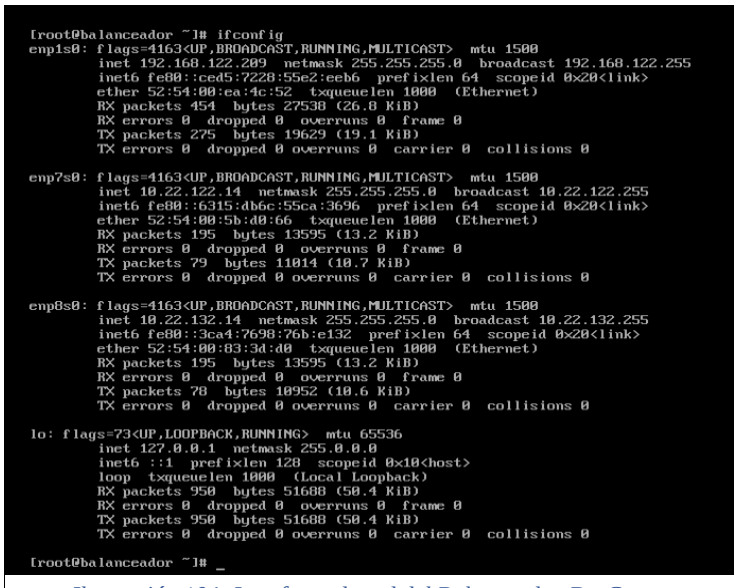

Ilustración 104: Interfaces de red del Balanceador-De-Carga

Anexo XXII: Interfaces de red del Balanceador-De-Carga

Las interfaces de red del Almacenamiento-iSCSI son:

```
[root@localhost ~]# ifconfig
enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.122.101  netmask 255.255.255.0  broadcast 192.168.122.255
        inet6 fe80::9663:9783:9d61:a8ae  prefixlen 64  scopeid 0x20<link>
        ether 52:54:00:34:2a:80  txqueuelen 1000  (Ethernet)
        RX packets 344  bytes 24079 (23.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 48  bytes 3857 (3.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp9s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.22.122.15  netmask 255.255.255.0  broadcast 10.22.122.255
        inet6 fe80::3700:48a8:99ab:e961  prefixlen 64  scopeid 0x20<link>
        ether 52:54:00:f4:ed:7d  txqueuelen 1000  (Ethernet)
        RX packets 313  bytes 20411 (19.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 93  bytes 11718 (11.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@localhost ~]# _
```

Ilustración 105: Interfaces de red del Almacenamiento-iSCSI

Anexo XXIII: Comprobación del correcto funcionamiento de Apache

Para comprobar que el servidor web Apache se ha instalado y se ha configurado de manera exitosa tanto en el Nodo1 como en el Nodo2, se puede acceder mediante las direcciones IP del Nodo1 y del Nodo2, a la página inicial de Apache desde el navegador "*Firefox*" del sistema host anfitrión.

En primer lugar, se accederá mediante la dirección IP correspondiente a la interfaz conectada la red aislada de Control del Nodo1 al servidor web Apache (Ver ***Ilustración 106***):

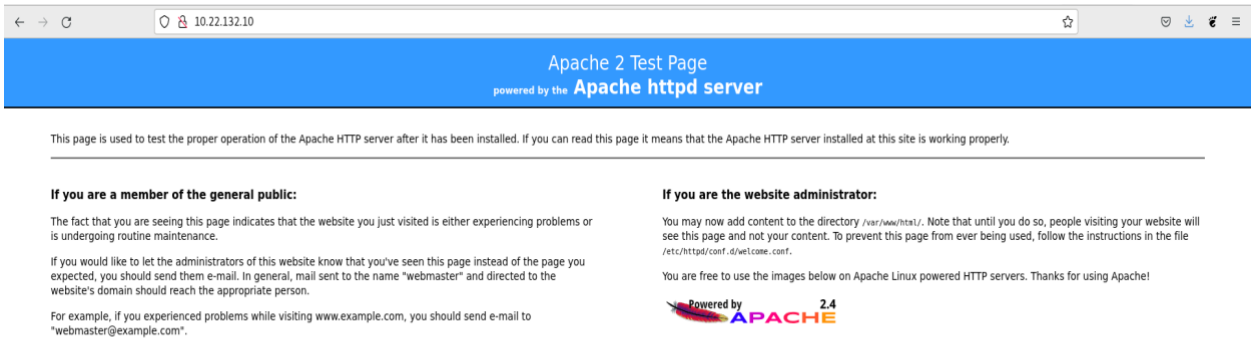

Ilustración 106: Comprobación del correcto funcionamiento de Apache en el Nodo1

En segundo lugar, se accederá mediante la dirección IP correspondiente a la interfaz conectada la red aislada de Control del Nodo2 al servidor web Apache (Ver ***Ilustración 107***):

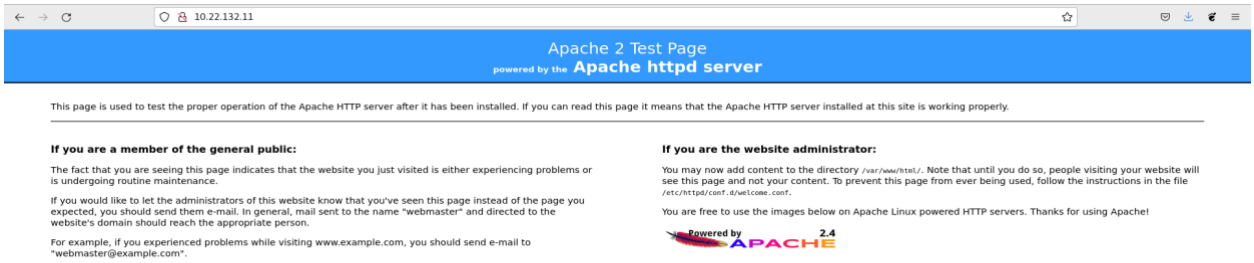

Ilustración 107: Comprobación del correcto funcionamiento de Apache en el Nodo2

Como se puede observar, el servidor web Apache funciona correctamente tanto en el Nodo1 como en el Nodo2.

Anexo XXIV: Comprobación del balanceo de carga y la alta disponibilidad del servidor *Apache*

A través de la dirección IP cliente **192.168.122.209** configurada en el servicio *HAProxy*, se podrá acceder al servidor web Apache. Para verificar que este servicio se está ofreciendo con balanceo de carga, se accederá desde el

navegador “Firefox” del host anfitrión, a través de la dirección IP cliente (192.168.122.209), al servidor web Apache (Ver ***Ilustración 108***):

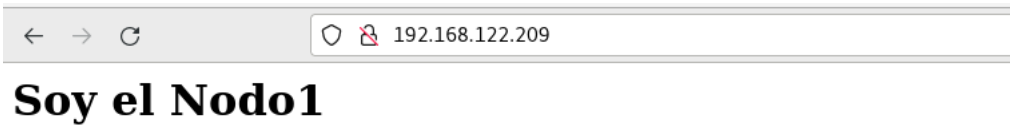

Ilustración 108: Servicio Apache ofrecido por el Nodo1

Como se puede observar en la anterior figura, el servicio *Apache* está siendo ofrecido en este momento por el Nodo1, pero si se recarga la ventana del navegador, se puede observar como ahora es el Nodo2 quién ofrecer el servicio Apache, consiguiendo de esta manera el equilibrio de carga deseado (Ver ***Ilustración 109***):

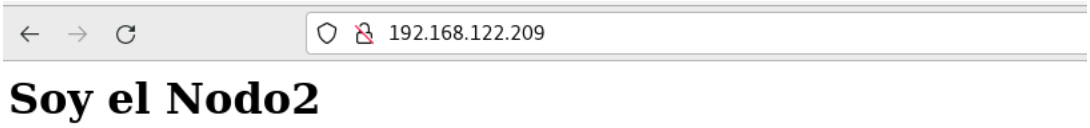

Ilustración 109: Servicio Apache ofrecido por el Nodo2

Además, para verificar que el servicio Apache se está ofreciendo también en alta disponibilidad, se puede parar el servicio *Apache* en uno de los dos nodos que lo ofrecen, por ejemplo, en el Nodo1:

```
$ systemctl stop httpd
```

Ahora, con el servicio Apache parado en el Nodo1, se accederá de nuevo desde el host anfitrión con la dirección **192.168.122.209** al servidor Apache y se puede comprobar que el servicio sigue activo en todo momento, pero esta vez solamente es ofrecido por el Nodo2, ya que el Nodo1 está inoperativo (Ver ***Ilustración 110***).

Ilustración 110: Servicio Apache ofrecido solamente por el Nodo2

Anexo XXV: Pasos para la creación del mecanismo de fencing para aislar a los nodos con fallas del clúster

Este mecanismo de **fencing** está basado en el host anfitrión, por lo que en primer lugar se deben instalar diferentes paquetes software para la configuración de este mecanismo desde el sistema anfitrión:

```
$ dnf -y install fence-virt fence-virtd fence-virtd-multicast
fence-virtd-libvirt
```

Además, como se está usando como puerto predeterminado el **1229** (*fence_virtd*), hay que permitir este puerto TCP en el firewall. Para ello, en primer lugar, se deberán analizar las zonas activas del cortafuegos del sistema anfitrión (Ver ***Ilustración 111***):

Ilustración 111: Zonas activas en el firewall del host anfitrión

Como se puede observar en la imagen anterior, hay solamente una zona activa en el firewall del host anfitrión denominada "***libvirt***", por lo que se procederá a abrir el puerto 1229 en la zona *libvirt* del cortafuegos del host anfitrión tanto para TCP como para UDP (Ver ***Ilustración 112***):

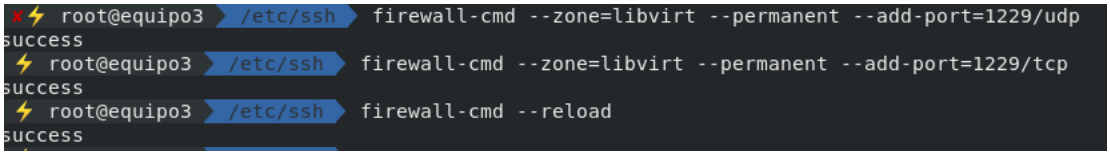

Ilustración 112: Añadir puerto 1229 a la zona libvirt del firewall del host anfitrión

Acto seguido, para configurar este mecanismo de *fencing* se deberá generar un fichero de configuración denominado "**fence_virt.conf**" que se encuentra bajo el directorio "*/etc*". Este archivo contiene varios parámetros de configuración tales como: dirección *multicast*, puerto *multicast*, un fichero con una clave compartida, etc. Para la creación de este fichero, se puede hacer uso de la siguiente orden desde el host anfitrión:

```
$ fence_virtd -c
```

Tras la ejecución de la orden anterior, se solicitarán ciertos valores por pantalla, de los cuáles se podrán seleccionar sus opciones por defecto excepto cuando se solicite el nombre de la interfaz. En esta opción se deberá indicar la interfaz de red del host anfitrión que corresponda a la red de Control creada anteriormente en KVM (En este caso, **virbr2**). Una vez finalizada la ejecución del comando, se habrá creado el fichero ***"/etc/fence_virt.conf***" y se podrá mostrar su contenido (Ver ***Ilustración 113***):

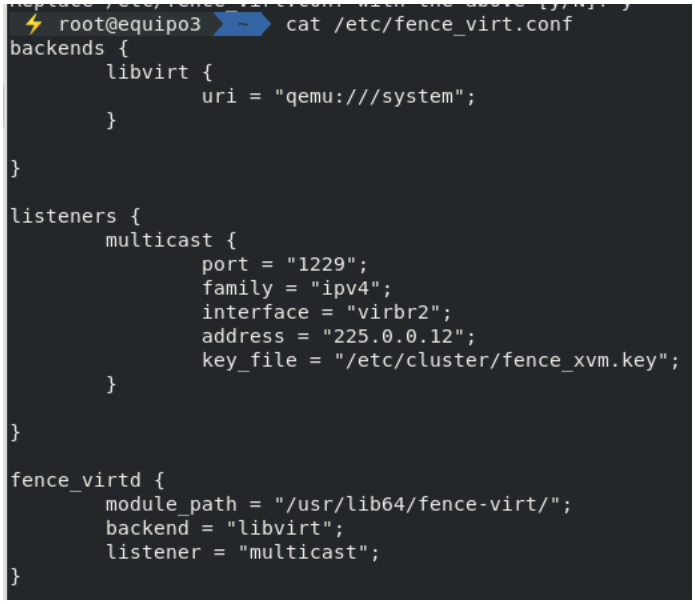

Ilustración 113: Contenido del fichero de configuración del mecanismo de fencing

A continuación, se deberá crear un fichero que contenga una clave compartida en su interior para la configuración del mecanismo de *fencing*. Por defecto, el directorio "***/etc/cluster"*** no existe en el host anfitrión, por lo que se debe crear de forma manual (*mkdir*). Tras crear el fichero y asignarle los permisos adecuados, se puede generar la clave compartida mediante el comando "***dd***". Todos estos pasos están recogidos en la siguiente imagen:

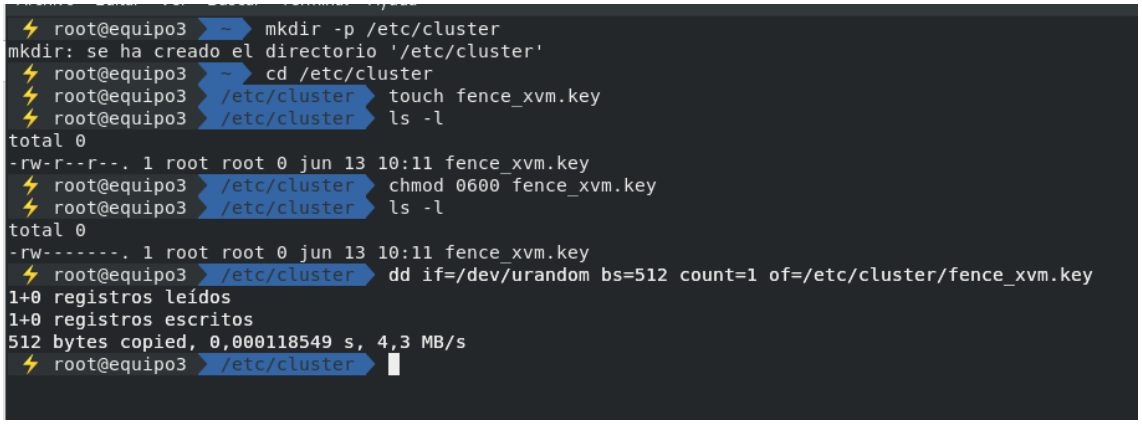

Ilustración 114: Creación del fichero que contiene la clave compartida del mecanismo de aislamiento

Una vez creada y configurada la clave compartida, se debe iniciar y arrancar en el inicio del sistema anfitrión, el servicio ***fence_virtd***. Para ello, se debe ejecutar en la máquina host:

```
$ systemctl enable fence_virtd
```

```
$ systemctl start fence_virtd
```

Y comprobar su estado mediante la orden "**systemctl":**

```
root@equipo3 /etc/cluster systemctl status fence_virtd
● fence_virtd.service - Fence-Virt system host daemon
     Loaded: loaded (/usr/lib/systemd/system/fence_virtd.service; enabled; vendor preset: disabled)
     Active: active (running) since Mon 2022-06-13 10:13:46 WEST; 12s ago
       Docs: man:fence_virtd(8)
             man:fence_virt.con(5)
   Main PID: 12789 (fence_virtd)
      Tasks: 1 (limit: 100126)
     Memory: 2.5M
     CGroup: /system.slice/fence_virtd.service
             └─12789 /usr/sbin/fence_virtd -w

jun 13 10:13:46 equipo3.tfg.com systemd[1]: Starting Fence-Virt system host daemon...
jun 13 10:13:46 equipo3.tfg.com fence_virtd[12789]: fence_virtd starting.  Listener: multicast  Backend:>
jun 13 10:13:46 equipo3.tfg.com systemd[1]: Started Fence-Virt system host daemon.
lines 1-14/14 (END)
```

Ilustración 115: Estado del demonio fence_virtd

Y también, comprobar la conectividad ***multicast***, en dónde se deben mostrar todas las máquinas virtuales de KVM y su estado (*on – off*), como se observa en la siguiente ilustración:

```
root@equipo3 /etc/cluster fence_xvm -o list
Balanceador-De-Carga            6d6f9957-c5c1-409d-af19-629a638086fd on
fedora35                        f52bc323-bc9e-4fb5-a064-e635c3e71d19 off
Maquina_Red_Puente              98f28989-1959-4480-bea3-1e9ac696de89 off
MaquinaBaseTFG                  a519667a-7c97-460e-8e1a-e2e0f5227826 off
Nodo1-Servicio                  db1aaf20-0a5d-44c9-bceb-b977d379bce7 on
Nodo2-Servicio                  bd83f534-d7e3-43b0-a767-86e132b2eecc on
Nodo3-Almacenamiento            11c6cde9-8344-4329-96f4-6097c175f827 on
Nodo4-Almacenamiento            2407c0a2-5c52-469f-8ee4-ecda4896e1b4 on
Prueba_clonacion                ec3a24dd-4ff4-4794-949c-55e1578c3a22 off
Prueba_copiando_ficheros        e027a5bc-5b48-4c68-a470-f70079a3fb81 off
Prueba_virt_clone               28a56d15-a366-49cb-87fe-3519b7c6af9f off
Prueba_virt_install             7bf04820-fb92-48e9-95bd-9c58adc323dd off
PruebaVirtManager               1c83a479-32dc-4eec-929a-35d3b164494f off
Simulacion_Servidor_NFS         cfa14003-bd3b-4903-8fff-6dba7edea2e4 off
root@equipo3 /etc/cluster
```

Ilustración 116: Comprobación de la conectividad multicast

Una vez configurado el mecanismo de ***fencing*** en el host anfitrión, se deberá realizar las configuraciones necesarias de este mecanismo en los nodos del clúster. En primer lugar, de deberán instalar los paquetes software *fence-virt* y *fence-virtd*, en todos los nodos del clúster (Nodo1, Nodo2, Nodo3, Nodo4 y Balanaceador-De-Carga).

```
$ dnf -y install fence-virt fence-virtd
```

Acto seguido, se debe configurar el cortafuegos de cada nodo del clúster (Nodos 1, 2, 3, 4 y de Balanceo) para permitir el tráfico con el host anfitrión a través de la red de Control (10.22.132.1) y del puerto 1229 (*fence_virtd*):

```
$ firewall-cmd --permanent --add-rich-rule=’rule family=”ipv4” 
source address=”10.22.132.1” port port=”1229” protocol=”tcp” 
accept’
```

```
$ firewall-cmd --reload
```

Tras esto, se procederá a crear el directorio administrativo que se ha creado anteriormente en el host anfitrión “**/etc/cluster**”. Este directorio se deberá crear en los cinco nodos del clúster:

```
$ mkdir -p /etc/cluster
```

Una vez creado el directorio anterior en los cinco nodos del clúster, se procederá a copiar la clave compartida creada anteriormente en el host anfitrión (*fence_xvm.key*) mediante el comando “**dd**”, al directorio *“/etc/cluster*” de cada nodo. Para ello, se podrá utilizar, por ejemplo, la orden “***scp***”.

```
$ scp root@10.22.132.1:/etc/cluster/fence_xvm.key /etc/cluster
```

Tras copiar la clave compartida (fence_xvm.key) del host anfitrión en el directorio ***"/etc/cluster"*** de cada uno de los cinco nodos, es muy importante verificar la conectividad multicast nuevamente, pero ahora desde cada uno de los nodos del clúster:

```
[root@nodo1 ~]# fence_xvm -o list
Balanceador-De-Carga            6d6f9957-c5c1-409d-af19-629a638086fd on
fedora35                        f52bc323-bc9e-4fb5-a064-e635c3e71d19 off
Maquina_Red_Puente              98f28989-1959-4480-bea3-1e9ac696de89 off
MaquinaBaseTFG                  a519667a-7c97-460e-8e1a-e2e0f5227826 off
Nodo1-Servicio                  db1aaf20-0a5d-44c9-bceb-b977d379bce7 on
Nodo2-Servicio                  bd83f534-d7e3-43b0-a767-86e132b2eecc on
Nodo3-Almacenamiento            11c6cde9-8344-4329-96f4-6097c175f827 on
Nodo4-Almacenamiento            2407c0a2-5c52-469f-8ee4-ecda4896e1b4 on
Prueba_clonacion                ec3a24dd-4ff4-4794-949c-55e1578c3a22 off
Prueba_copiando_ficheros        e027a5bc-5b48-4c68-a470-f70079a3fb81 off
Prueba_virt_clone               28a56d15-a366-49cb-87fe-3519b7c6af9f off
Prueba_virt_install             7bf04820-fb92-48e9-95bd-9c58adc323dd off
PruebaVirtManager               1c83a479-32dc-4eec-929a-35d3b164494f off
Simulacion_Servidor_NFS         cfa14003-bd3b-4903-8fff-6dba7edea2e4 off
[root@nodo1 ~]# _
```

Ilustración 117: Conectividad multicast del Nodo1

Por último, se debe crear el mecanismo de *fencing* como un recurso del clúster. Este recurso será de tipo "**fence_xvm**", hará uso de la clave compartida y recibirá el nombre de "**xvmfence**". Para la creación de este recurso de *fencing* se usará la orden "***pcs stonith***". Por lo tanto, se debe ejecutar desde cualquier nodo del clúster (solo uno de ellos, en este caso desde el Nodo1) la orden:

```
$ pcs stonith create xvmfence fence_xvm
key_file=/etc/cluster/fence_xvm.key
```

Tras la creación del mecanismo de fencing como recurso del clúster, se puede ver el estado de este recurso haciendo uso de la orden "**pcs stonith**" (Ver ***Ilustración 118***):

```
[root@nodo1 ~]# pcs stonith
  * xvmfence    (stonith:fence_xvm):    Started nodo1.tfg.com
```

Ilustración 118: Estado del recurso de aislamiento de nodos del clúster

También, se puede verificar la alta disponibilidad de este recurso de fencing. Para ello, se observará el estado del clúster con la orden "**pcs status**" (Ver ***Ilustración 119***):

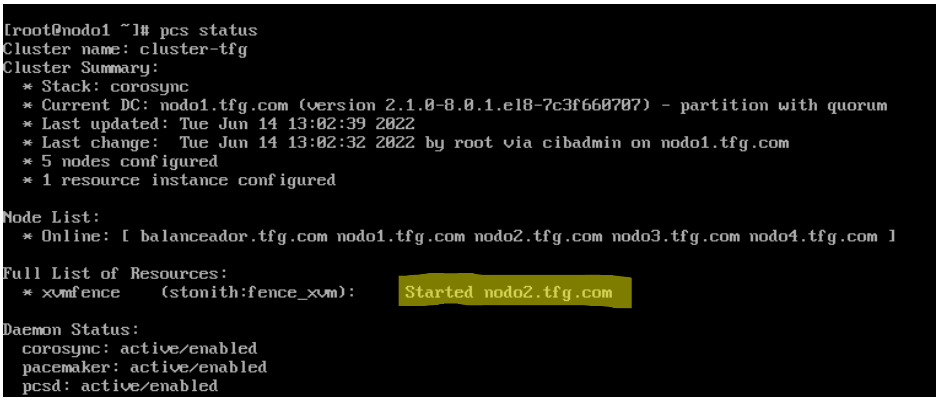

Ilustración 119: Estado del clúster con todos los nodos online

Como se puede observar en la ilustración anterior, el mecanismo de fencing está siendo ofrecido por el Nodo2. Acto seguido, se aislará del clúster al Nodo2 (nodo que estaba ofreciendo el servicio de *fencing*) mediante la orden: "**pcs node standby nodo2.tfg.com**". Una vez que se aísla al Nodo2 del clúster, el mecanismo de *fencing* seguirá siendo ofrecido, pero esta vez, cogerá el testigo alguno de los cuatro nodos restantes del clúster, proporcionando de esta manera alta disponibilidad. En la ***Ilustración 120,*** se puede apreciar el estado del clúster y se puede observar cómo es ahora el nodo Balanceador-De-Carga, el encargado de ofrecer el servicio, ya que el Nodo2 está Offline.

Ilustración 120: Estado del clúster con el Nodo2 aislado

Nota: La orden "**pcs node unstandby nodo2.tfg.com**" vuelve a poner "Online" al Nodo2.

Anexo XXVI: Pruebas del funcionamiento de la replicación síncrona y la alta disponibilidad del clúster de Galera

Primero que nada, se procederá a comprobar que los datos de las bases de datos se replican de un nodo del clúster de Galera al otro nodo de manera síncrona. Para ello, se creará mediante el cliente de MariaDB, una nueva base de datos desde el Nodo3 llamada "***prueba_replicacion***", tal y como se muestra en la siguiente ilustración:

```
[root@nodo3 my.cnf.d]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 14
Server version: 10.3.32-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE prueba_replicacion;
Query OK, 1 row affected (0.010 sec)

MariaDB [(none)]> _
```

Ilustración 121: Creación de una BD llamada "prueba_replicacion" en el Nodo3

Acto seguido, se accederá al Nodo4 y se hará uso del cliente de MariaDB para mostrar por pantalla el listado de bases de datos creadas, tal y como se observa en la siguiente ilustración:

```
[root@nodo4 my.cnf.d]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 11
Server version: 10.3.32-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| prueba_replicacion |
+--------------------+
4 rows in set (0.001 sec)

MariaDB [(none)]>
```

Ilustración 122: Listado de Bases de datos desde el Nodo4

Como se observa en la imagen anterior, en el listado de bases de datos del Nodo4, aparece la base de datos denominada "**prueba_replicacion**" creada anteriormente desde el Nodo3, ya que se ha replicado. Por lo tanto, la replicación síncrona de datos en el clúster de Galera funciona perfectamente.

Además, si paramos el servicio MariaDB en alguno de los nodos 3 o 4 (***systemctl stop mariadb***), la base de datos seguirá disponible en el nodo que esté operativo, ofreciendo de esta manera alta disponibilidad en la base de datos. Cuando el nodo que estaba inoperativo vuelva a la normalidad, se volverá a unir al clúster de Galera y podrá ofrecer de nuevo con los datos actualizados la base de datos.

Anexo XXVII: Pruebas del balanceo de carga en las peticiones a la base de datos

Para probar el funcionamiento del balanceo de carga en las peticiones a la base de datos, se procederá realizando una instalación segura de la base de datos MariaDB en los nodos: **Nodo3** y **Nodo4**, con el fin de establecer una contraseña para el usuario "**root**". Para ello, se hará uso del siguiente comando:

```
$ mysql_secure_installation
```

A continuación, se creará un usuario llamado "**prueba_balanceo**" identificado con la contraseña "**prueba_balanceo**" y con privilegios en la base de datos "**prueba_replicacion**". Todos estos pasos se realizarán, por ejemplo, desde el Nodo3 (Ver **Ilustración 123**).

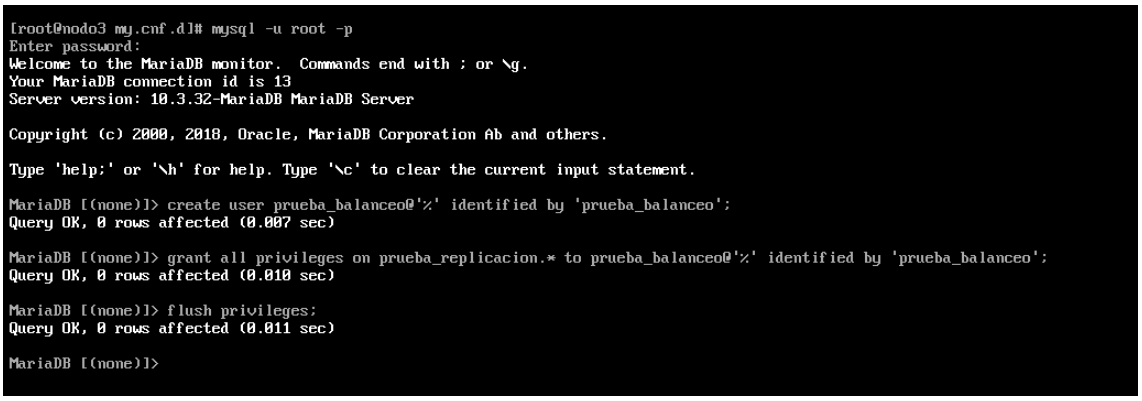

Ilustración 123: Creación del usuario "prueba_balanceo" con privilegios en la BD "prueba_replicacion"

Una vez creado este usuario con privilegios, se procederá a realizar una conexión desde el Nodo1 mediante el cliente de MariaDB para consultar el valor de la variable "**wsrep_node_name**", que hace referencia al nombre del nodo que está ofreciendo el servicio MariaDB. Para ello, se deberá previamente instalar MariaDB en el Nodo1 de la siguiente forma:

```
$ dnf -y install mariadb-server
```

Y habilitarlo en el arranque del sistema a la vez que iniciarlo:

```
$ systemctl enable –now mariadb
```

Una vez instalado e iniciado el servicio MariaDB en el Nodo1, se procederá a consultar el valor de la variable "**wsrep_node_name**" haciendo uso del usuario "**prueba_balanceo**" y de la dirección cliente **192.168.122.209** (configurada en el fichero haproxy.cfg), para saber qué nodo está ofreciendo el servicio MariaDB en cada momento (Ver ***Ilustración 124***):

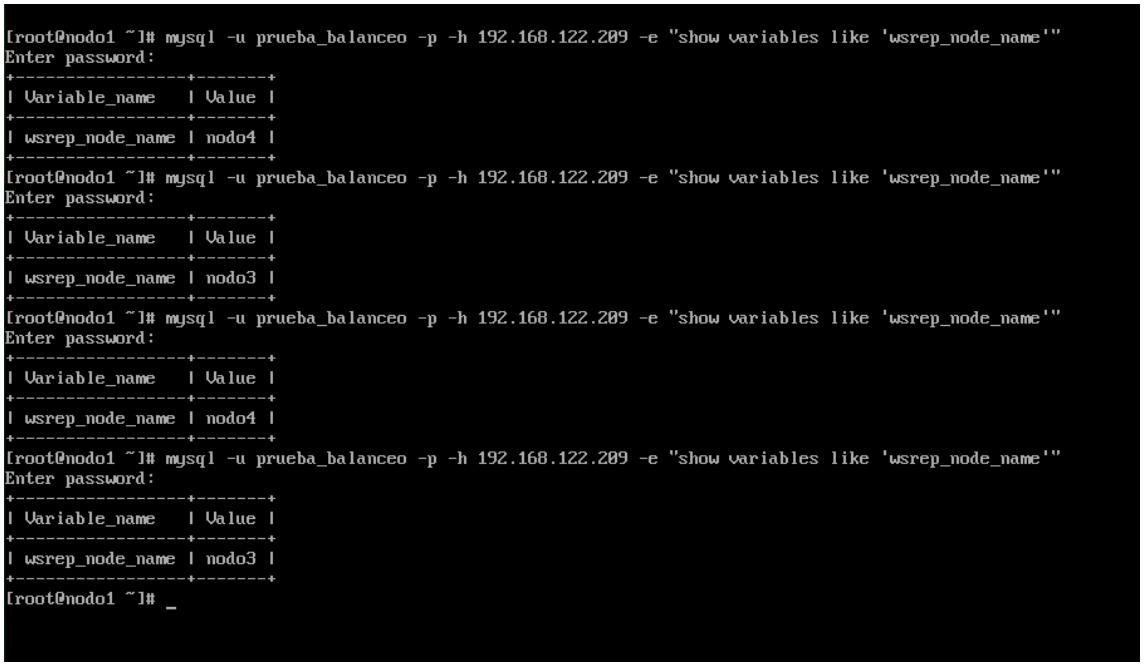

Ilustración 124: Comprobación del balanceo de carga en la base de datos

Como se puede observar en la imagen anterior, en cada consulta que se realiza mediante el cliente MariaDB del Nodo1 a la variable "**wsrep_node_name**", nos responde un nodo diferente, ya que en la primera consulta responde el Nodo4, en la segunda el nodo3, en la tercera el Nodo4 y en la cuarta otra vez el Nodo3. De esta manera, ahora las peticiones a la base de datos están distribuidas equitativamente gracias al balanceo de carga, evitando la sobrecarga de peticiones en un único servidor.

Si ahora paramos el servicio MariaDB en el Nodo4 (***systemctl stop mariadb***) y volvemos a consultar el valor de la variable "**wsrep_node_name**" desde el Nodo1, se puede observar como ahora solo responde el Nodo3, ya que el Nodo 4 está inoperativo. Además, también se puede observar que el tamaño del clúster de Galera es de "1", ya que el Nodo4 está inoperativo, pero el servicio de base de datos sigue activo en el Nodo3 gracias a la alta disponibilidad (Ver ***Ilustración 125***).

```
[root@nodo1 ~]# mysql -u prueba_balanceo -p -h 192.168.122.209 -e "show variables like 'wsrep_node_name'"
Enter password:
+-----------------+-------+
| Variable_name   | Value |
+-----------------+-------+
| wsrep_node_name | nodo3 |
+-----------------+-------+
[root@nodo1 ~]# mysql -u prueba_balanceo -p -h 192.168.122.209 -e "show variables like 'wsrep_node_name'"
Enter password:
+-----------------+-------+
| Variable_name   | Value |
+-----------------+-------+
| wsrep_node_name | nodo3 |
+-----------------+-------+
[root@nodo1 ~]# mysql -u prueba_balanceo -p -h 192.168.122.209 -e "show variables like 'wsrep_node_name'"
Enter password:
+-----------------+-------+
| Variable_name   | Value |
+-----------------+-------+
| wsrep_node_name | nodo3 |
+-----------------+-------+
[root@nodo1 ~]# mysql -u prueba_balanceo -p -h 192.168.122.209 -e "show status like 'wsrep_cluster_size'"
Enter password:
+--------------------+-------+
| Variable_name      | Value |
+--------------------+-------+
| wsrep_cluster_size | 1     |
+--------------------+-------+
[root@nodo1 ~]# _
```

Ilustración 125: Comprobación del balanceo de carga y la alta disponibilidad en la base de datos con el Nodo4 inoperativo

Anexo XXVIII: Pruebas del funcionamiento de la interfaz gráfica de estadísticas de HAProxy

Accediendo a través del navegador "Firefox" del host anfitrión, a la dirección http://192.168.122.209:9000/haproxy_stats y con el usuario y contraseña "haproxy", se podrá acceder a la interfaz gráfica de estadísticas del servicio HAProxy (Ver ***Ilustración 126***):

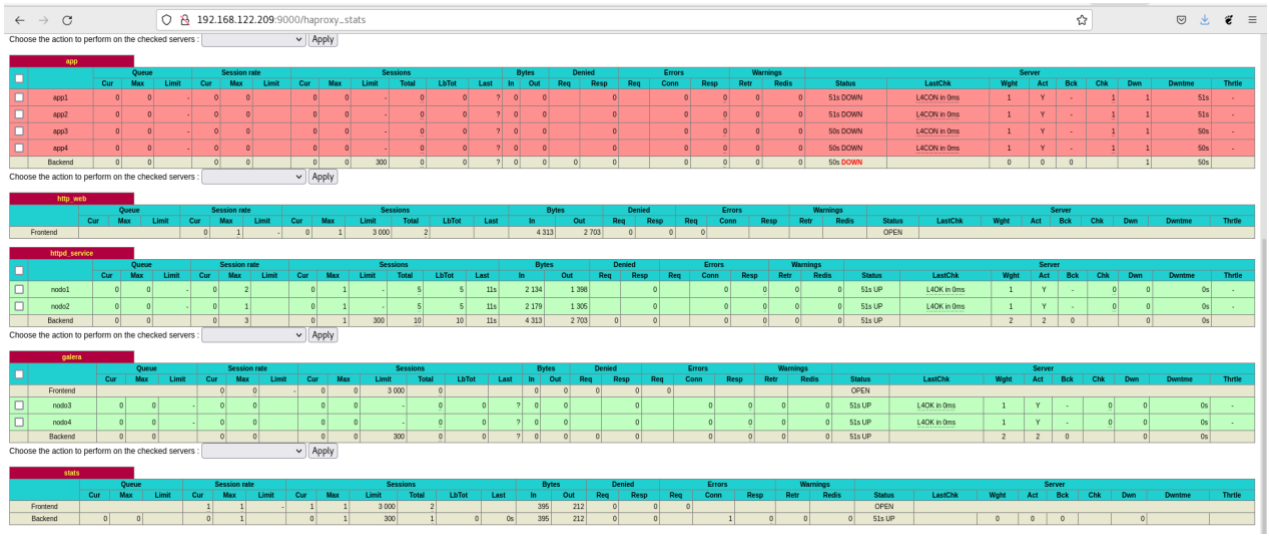

Ilustración 126: Interfaz gráfica de estadísticas de HAProxy

Para observar alguna estadística proporcionada por esta interfaz web, se procederá a parar el servicio "*mariadb*" en el Nodo4 (***systemctl stop mariadb***) y el servicio Apache en el Nodo1 (**systemctl stop httpd**). Tras parar el servicio MariaDB en el Nodo4 y el servicio Apache en el Nodo1 y volver a acceder a la interfaz gráfica de estadísticas desde el navegador del host anfitrión con la URL http://192.168.122.209:9000/haproxy_stats, se puede observar como ahora en la sección "galera", el Nodo4 se muestra en color "rojo" ya que está inoperativo y en la sección "httpd_service", el Nodo1 se muestra también en color "rojo" porque también está inoperativo (Ver ***Ilustración 127***).

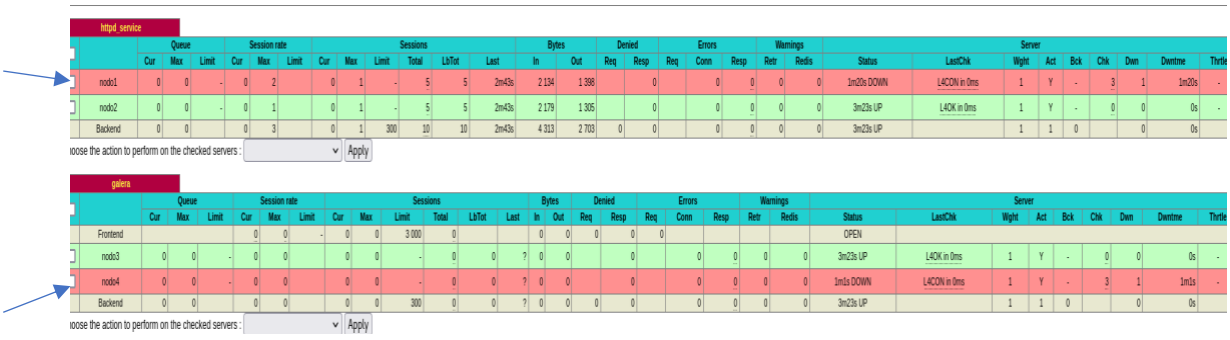

Ilustración 127: Interfaz gráfica de estadísticas de HAProxy tras la caída de dos nodos

Anexo XXIX: Configuraciones para el correcto funcionamiento de PHP

En primer lugar, se creará un nuevo fichero denominado "**info.php**" bajo el directorio "***/var/www/html***" de los Nodos 1 y 2, que muestre la información de la versión de PHP instala vía web. Para ello, se creará el fichero "***/var/www/html/info.php"*** en los nodos 1 y 2 con el siguiente contenido (Ver ***Ilustración 128***):

```
[root@nodo1 ~]# cat /var/www/html/info.php
<?php
phpinfo();
?>
```

Ilustración 128: Contenido del fichero info.php en los Nodos 1 y 2

Acto seguido, se podrá ver la información de la versión de PHP instalada en los Nodos 1 y 2, accediendo vía web a la dirección "**192.168.122.209/info.php**" desde el host anfitrión (Ver ***Ilustración 129***):

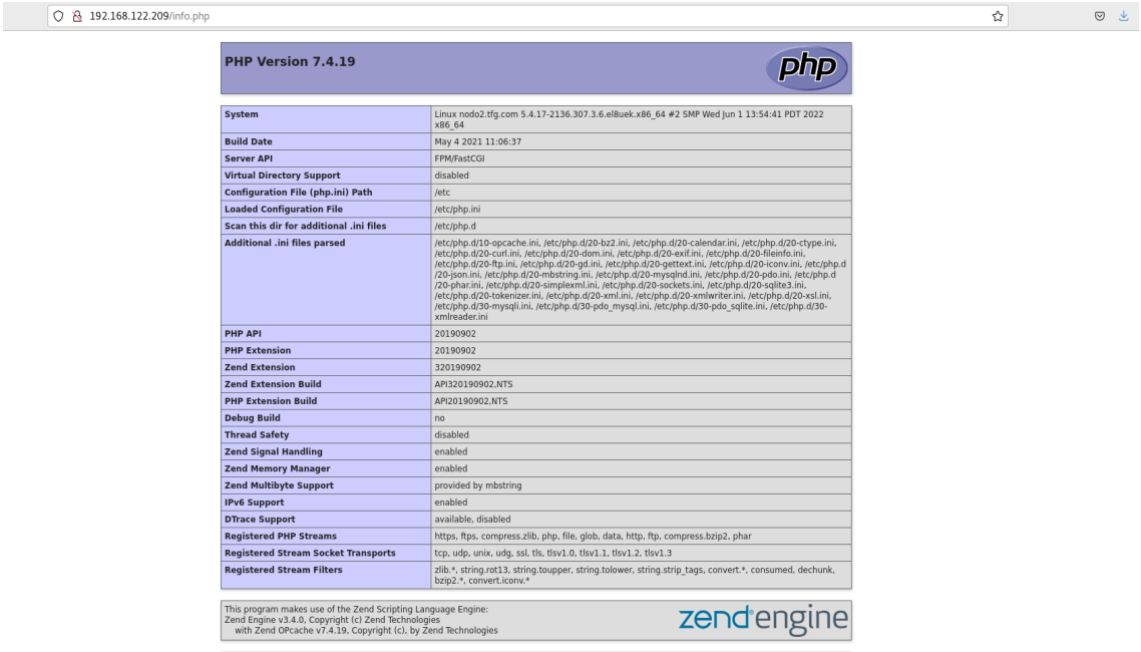

Ilustración 129: Información de PHP vía web

Por último, se modificarán cuatro parámetros relativos al fichero de configuración de PHP ***("/etc/php.ini")*** en los Nodos 1 y 2. En primer lugar, se modificarán los parámetros correspondientes al tratamiento de errores de PHP "***error_reporting***", "***display_errors***" y "***display_startup_errors***" en el fichero ***"/etc/php.ini***" de los Nodos 1 y 2 (Ver ***Ilustración 130***):

```
error_reporting = E_ALL

; This directive controls whether or not and where PHP will output errors,
; notices and warnings too. Error output is very useful during development, but
; it could be very dangerous in production environments. Depending on the code
; which is triggering the error, sensitive information could potentially leak
; out of your application such as database usernames and passwords or worse.
; For production environments, we recommend logging errors rather than
; sending them to STDOUT.
; Possible Values:
;   Off = Do not display any errors
;   stderr = Display errors to STDERR (affects only CGI/CLI binaries!)
;   On or stdout = Display errors to STDOUT
; Default Value: On
; Development Value: On
; Production Value: Off
; http://php.net/display-errors
display_errors = On

; The display of errors which occur during PHP's startup sequence are handled
; separately from display_errors. PHP's default behavior is to suppress those
; errors from clients. Turning the display of startup errors on can be useful in
; debugging configuration problems. We strongly recommend you
; set this to 'off' for production servers.
; Default Value: Off
; Development Value: On
; Production Value: Off
; http://php.net/display-startup-errors
display_startup_errors = On
```

Ilustración 130: Parámetros para el tratamiento de errores de PHP en los Nodos 1 y 2

Y acto seguido, se modificará la zona horaria de los Nodos 1 y 2, descomentando la línea "**date.timezone**" y añadiendo la zona horaria de Canarias. Para ello, se modificará el fichero ***"/etc/php.*ini"** en los Nodos 1 y 2 de la siguiente manera (Ver ***Ilustración 131***):

```
[Date]
; Defines the default timezone used by the date functions
; http://php.net/date.timezone
date.timezone = Atlantic/Canary_
```

Ilustración 131: Zona horaria del servidor en los Nodos 1 y 2

Anexo XXX: Pasos para la instalación de *Wordpress* desde el navegador

Para realizar la instalación de *Wordpress* desde el navegador "*Firefox*" del host anfitrión, se hará uso del servicio Apache (servidor web en el que está basado *Wordpress*). Cabe recordar que el servidor web Apache está configurado tanto en el Nodo 1 como en el Nodo 2, y además es ofrecido en alta disponibilidad y con balanceo de carga. Para evitar conflictos con respecto a la distribución de peticiones que se realiza en este servicio y que es proporcionado por el balanceador, se procederá a realizar la instalación de *Wordpress* con tan solo uno de los dos nodos que ofrece el servicio *Apache* operativo, consiguiendo de esta manera que en cada paso de la configuración de *Wordpress*, el servicio Apache esté proporcionado por el mismo nodo. Para ello, se procederá a parar el servicio Apache de manera momentánea en el Nodo2:

```
$ systemctl stop httpd
```

Una vez parado el servicio Apache en el Nodo2 (en este caso), se procederá con la instalación del CMS *Wordpress* en el navegador del sistema anfitrión. Para ello, se accederá mediante Firefox a la URL "***192.168.122.209/wp-admin/setup.config.php***".

En primer lugar, se debe seleccionar el idioma de *Wordpress* de la lista ofrecida (en la que se elegirá el "Español"):

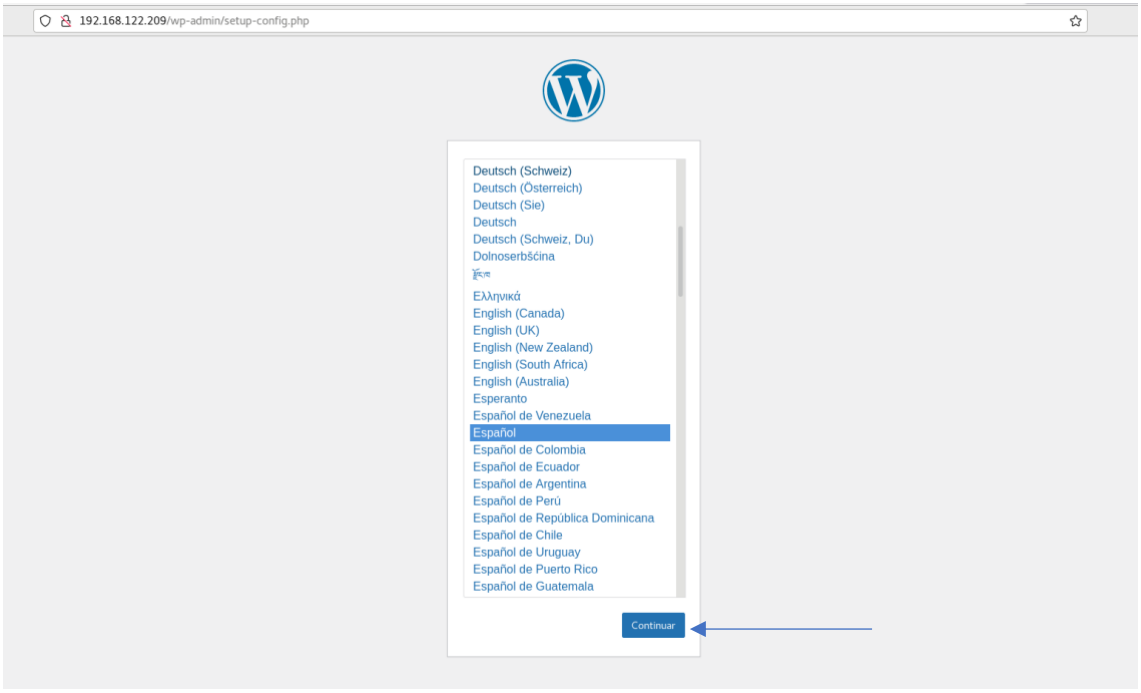

Ilustración 132: Idioma de Wordpress

Tras hacer clic en el botón "**Continuar**", se mostrará la página de bienvenida de *Wordpress*, en la que se debe seleccionar "**¡Vamos a ello!"**

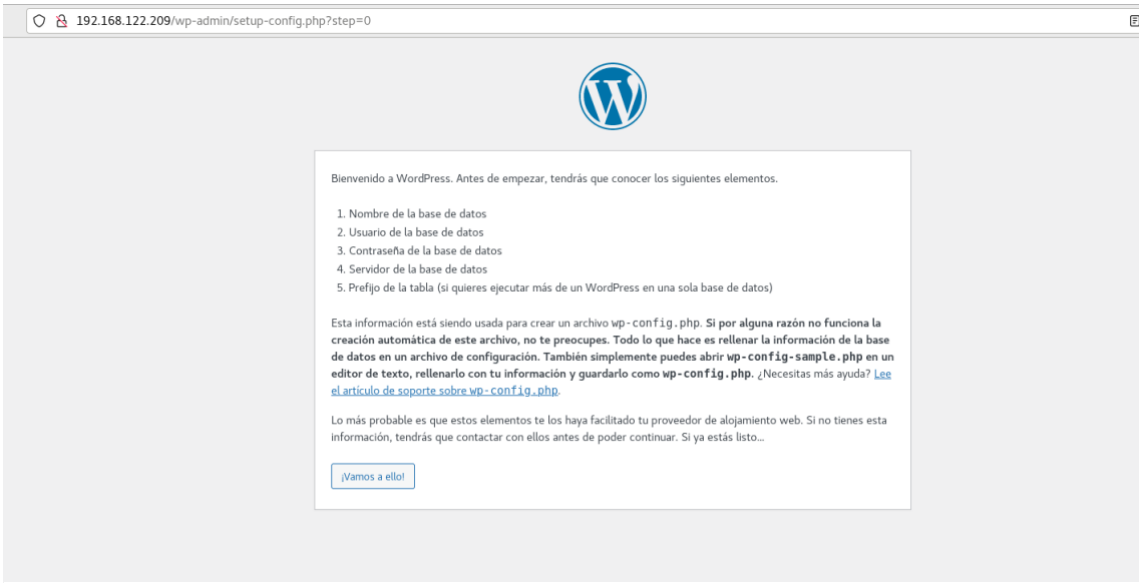

Ilustración 133: Página de bienvenida de Wordpress

El siguiente paso es muy importante, ya que se deben especificar los datos de la base de datos que va a usar *Wordpress*. Por lo que se debe indicar el nombre de la base de datos (**wordpress**), el usuario de la base de datos (**wordpress**), la contraseña de este usuario (**wordpress**) y la dirección IP del servidor de base de datos (**192.168.122.209**).

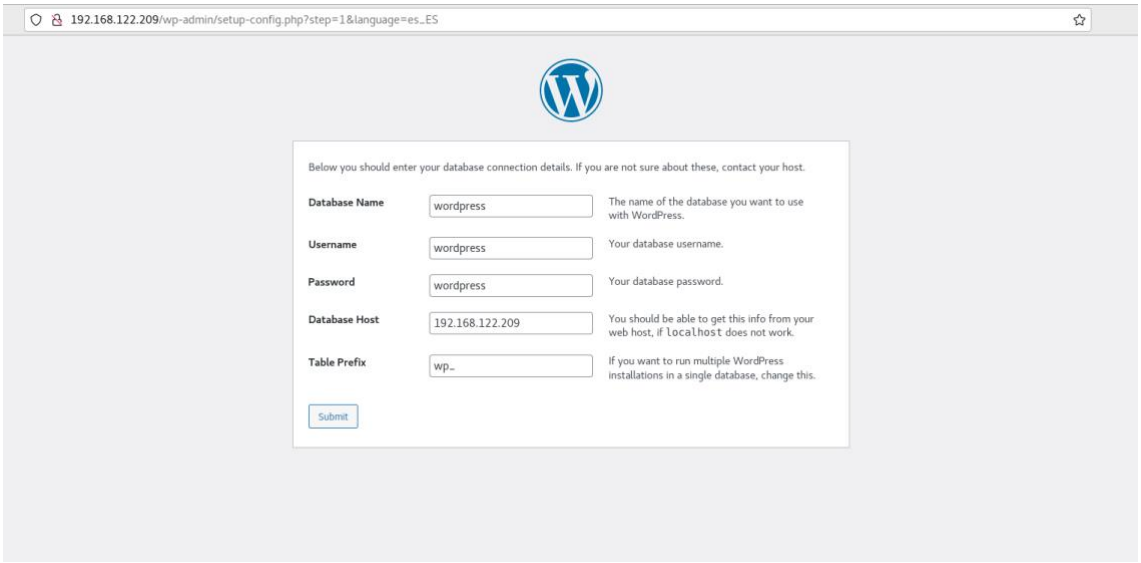

Ilustración 134: Parámetros de configuración de la Base de datos de Wordpress

Tras rellenar los datos de la base de datos, se ha concluido la primera parte de la configuración, por lo que se debe hacer clic en el botón "**Ejecutar la instalación**" para seguir con la configuración de Wordpress en el navegador.

Ilustración 135: Ejecutar la instalación de Wordpress

En el siguiente paso, se deben especificar los parámetros de inicio de sesión de *Wordpress*, así como el nombre del sitio web que se va a ofrecer (TFG, en este caso)

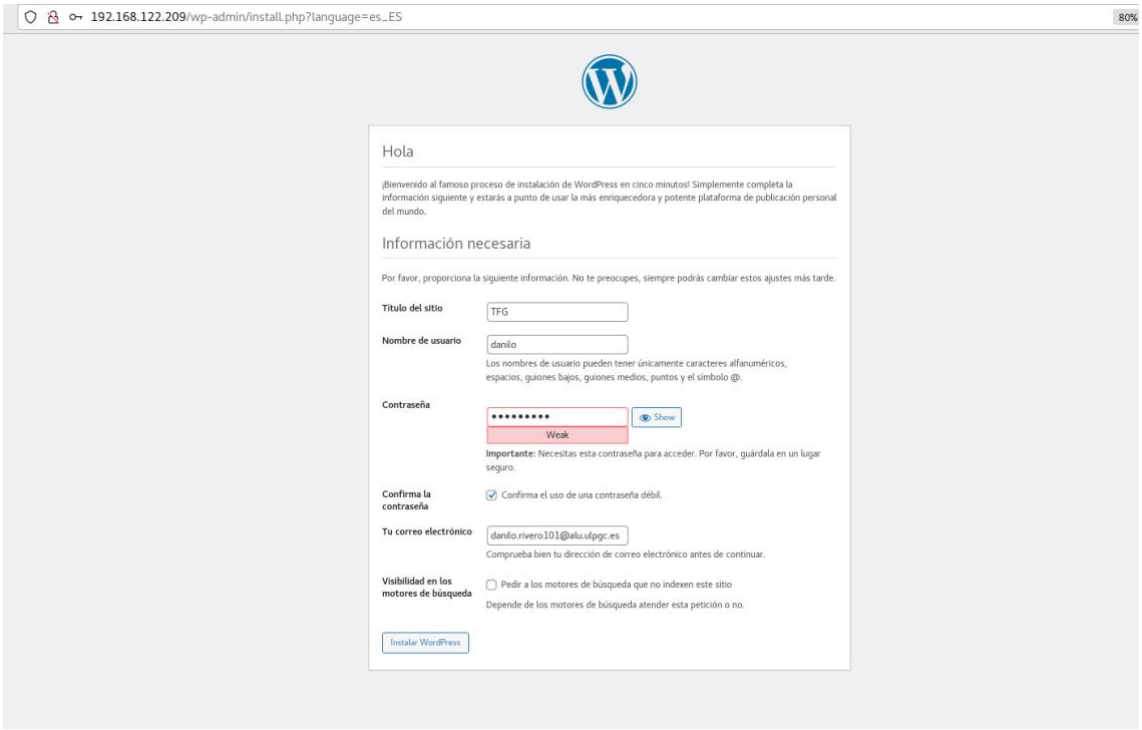

Ilustración 136: Parámetros del sitio web y de inicio de sesión de Wordpress

Una vez que proporcionado los parámetros anteriores, ya se habrá finalizado con la instalación de *Wordpress* desde el navegador y ya se podrá iniciar sesión en el CMS.

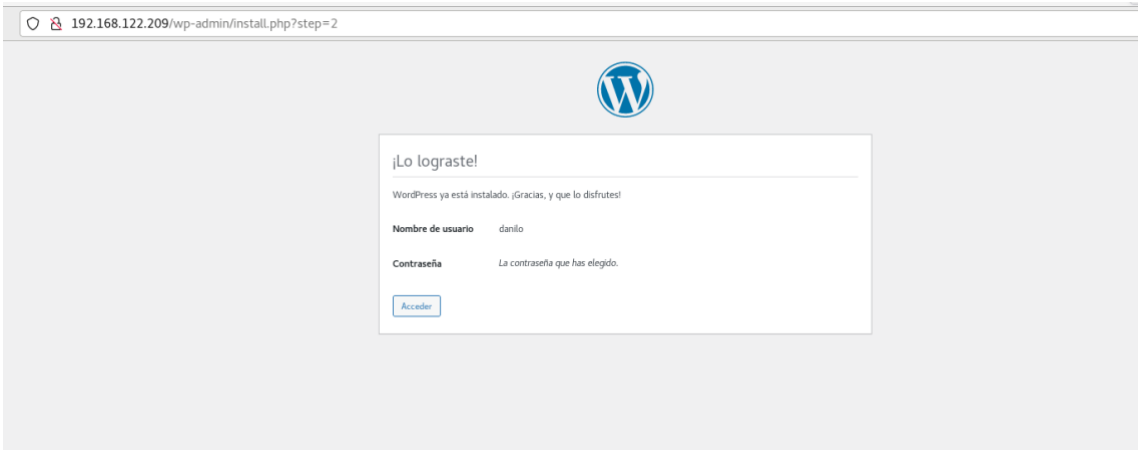

Ilustración 137: Instalación de Wordpress finalizada

Tras hacer clic en el botón "**Acceder**" de la imagen anterior, se redireccionará a la página de "*login*" de *Wordpress*, donde se deben especificar las credenciales especificadas en pasos anteriores (Ver **Ilustración 138**):

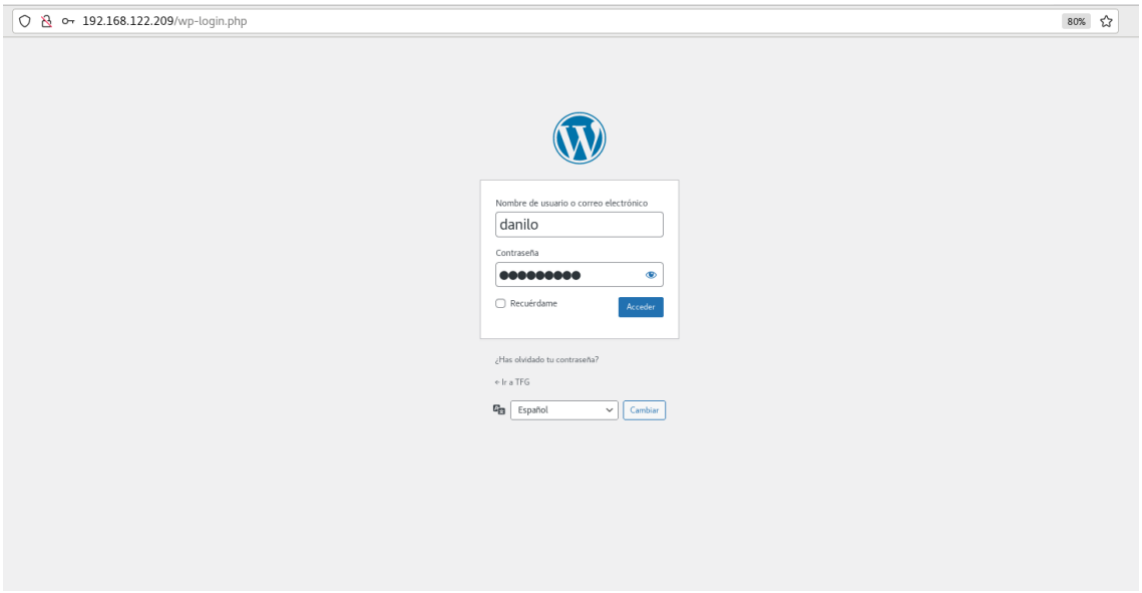

Ilustración 138: Página de Login de Wordpress

Tras el inicio de sesión, se tendrá acceso al panel de administración de *Wordpress* como se muestra en la imagen siguiente:

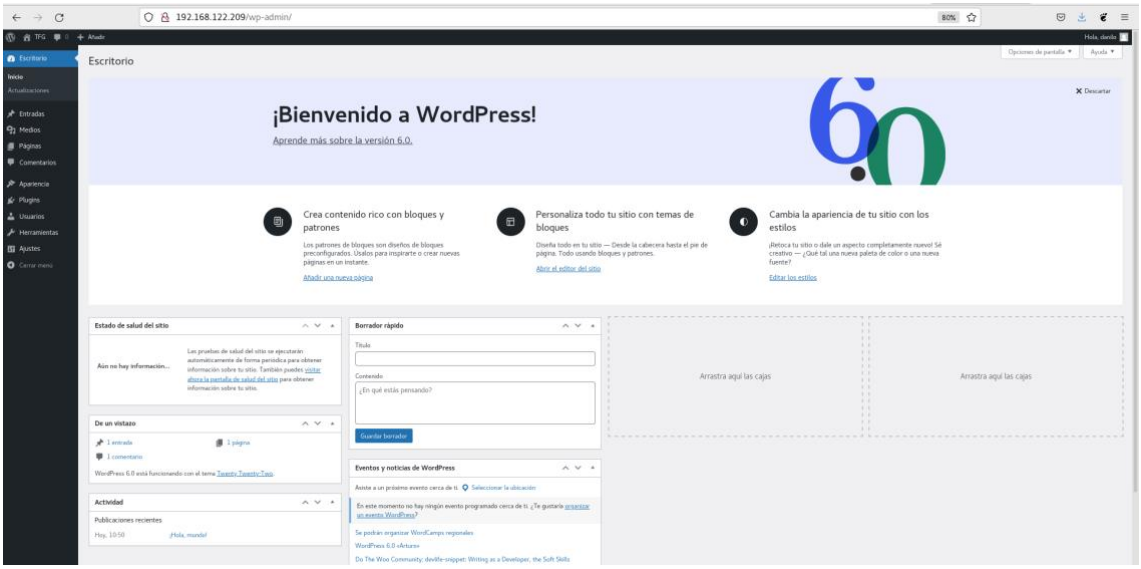

Ilustración 139: Panel de administración de Wordpress

Por último, ya se puede volver a iniciar el servicio Apache en el Nodo2 (***systemctl start httpd***).

Anexo XXXI: Verificación del servicio web *Wordpress* basado en Apache en alta disponibilidad y con balanceo de carga

Si se accede desde el navegador del host anfitrión a la dirección IP cliente ***192.168.122.209***, se puede observar el sitio web creado en *Wordpress*. Además, este sitio web basado en Apache, está ofrecido en alta disponibilidad y con balanceo de cargo. En la siguiente ilustración, se observa que el servicio web de Wordpress está siendo ofrecido por el Nodo1 (Ver ***Ilustración 140***):

Ilustración 140: Sitio web de Wordpress ofrecido por el Nodo1

Al recargar la página desde el navegador del host anfitrión, se puede observar que ahora es el Nodo2 quién ofrece el servicio web Wordpress, ya que las peticiones de los clientes a este servicio web se distribuyen mediante el equilibrador de carga (Balanceador-De-Carga) y el algoritmo roundrobin (Ver ***Ilustración 141***):

Ilustración 141: Sitio web de Wordpress ofrecido por el Nodo2

Cabe recordar, que para que *Wordpress* funcione, se hace uso de una base de datos. Esta base de datos se encuentra ofrecida en alta disponibilidad y con balanceo de carga por los Nodos 3 y 4, ya que, si uno de los nodos 3 o 4 no está operativo, la base de datos sigue operativa por uno de ellos. Además, si los Nodos 3 y 4 que conforman el clúster de Galera y que son los encargados de ofrecer la base de datos denominada "**wordpress**", están ambos operativos, entonces las peticiones a la base de datos se distribuyen entre el Nodo3 y el Nodo4, evitando la sobrecarga en los servidores (máquinas virtuales).

Por último, mencionar que si se para el servicio Apache en el Nodo1 o en el Nodo2 (systemctl stop httpd), el servicio Wordpress seguirá operativo por el nodo que tenga el servicio arrancado, verificando la alta disponibilidad en el servicio ofrecido.

Anexo XXXII: Fichero de configuración del clúster OCFS2

```
cluster:
        heartbeat_mode = local
        node_count = 5
        name = clusterOCFS2

node:
        number = 0
        cluster = clusterOCFS2
        ip_port = 7777
        ip_address = 10.22.122.10
        name = nodo1.tfg.com

node:
        number = 1
        cluster = clusterOCFS2
        ip_port = 7777
        ip_address = 10.22.122.11
        name = nodo2.tfg.com

node:
        number = 2
        cluster = clusterOCFS2
        ip_port = 7777
        ip_address = 10.22.122.12
        name = nodo3.tfg.com

node:
        number = 3
        cluster = clusterOCFS2
        ip_port = 7777
        ip_address = 10.22.122.13
        name = nodo4.tfg.com

node:
        number = 4
        cluster = clusterOCFS2
        ip_port = 7777
        ip_address = 10.22.122.14
        name = balanceador.tfg.com
```

Ilustración 142: Fichero de la configuración del clúster de OCFS2

Anexo XXXIII: Pasos para lograr acceso concurrente y alta disponibilidad en los ficheros de configuración de *Apache*

En primer lugar, se creará un volumen lógico de 4GB llamado "**lv_apache_server**" que haga uso del espacio disponible en el grupo de volúmenes "**vg_almacenamiento_compartido**". Desde el Nodo1, se creará este volumen lógico de la siguiente manera:

```
$ lvcreate -n lv_apache_server -L 4G vg_almacenamiento_compartido
```

Una vez creado el volumen lógico anterior en el que se guardarán los ficheros de configuración de Apache, se procederá a crear sobre él, un sistema de ficheros OCFS2 para otorgar acceso concurrente y alta disponibilidad a sus archivos.

```
$ mkfs.ocfs2 -N 2 /dev/mapper/vg_almacenamiento_compartido-
lv_apache_server
```

Nota: La opción "**-N**", indica el número de nodos que tienen acceso simultáneo al volumen.

Tras el paso anterior, es muy recomendable reiniciar todos los nodos del clúster OCFS2 (*reboot*), para que la configuración del volumen lógico recién creado se aplique en todos los nodos del clúster. Una vez arrancados los nodos, se debe verificar con la orden "*lsblk*" que el volumen lógico para Apache y su formato, aparecen correctamente en todos los nodos, ya que este volumen lógico está establecido sobre el almacenamiento compartido iSCSI.

Lo siguiente que se realizará, será la creación del directorio "/apache" en los Nodos 1 y 2, que albergará todos los ficheros de configuración del servidor web. Para ello:

```
$ mkdir -p /apache
```

Una vez creado el directorio para Apache en los nodos 1 y 2, se debe montar el volumen lógico correspondiente a Apache tanto en el Nodo1 como en el Nodo2. Además, se debe montar de tal manera que se monte automáticamente en el arranque de ambos nodos. Para ello, se debe modificar el fichero ***"/etc/fstab***" de los Nodos 1 y 2, añadiendo al final del mismo la siguiente línea:

```
/dev/mapper/vg_almacenamiento_compartido-lv_apache_server /apache ocfs2 _netdev 0 0
```

Nota: Se usa la opción "*_**netdev***", porque el sistema de archivos reside en un dispositivo que requiere acceso a la red.

Tras modificar el fichero "**fstab**" de los Nodos 1 y 2, se debe ejecutar la orden "*mount -a*" en los Nodos 1 y 2, para que se monte todo lo que se ha establecido en el "fstab".

Tras realizar todos los pasos anteriores, se ha montado el volumen lógico para los ficheros de configuración de *Apache*, en el directorio "**/apache**" de los nodos 1 y 2. Por lo tanto, llegados a este punto, se tiene acceso concurrente y síncrono en el directorio "/apache" de los Nodos 1 y 2, proporcionando de esta manera alta disponibilidad en el almacenamiento compartido.

Lo siguiente que se debe hacer, es crear los directorios "**httpd**" y "**www**" correspondientes al servicio Apache en el directorio "/apache" de los Nodos 1 y 2. Para la creación de estos dos directorios, tan solo hará falta crearlos en uno de los dos nodos (Nodo1 o Nodo2), ya que ahora, el directorio "/apache" dispone de acceso concurrente y síncrono, garantizando que los datos creados en este directorio, en uno de los nodos, se replican automáticamente en el otro nodo (alta disponibilidad). Para ello, se procederá creando estos dos directorios desde el Nodo1:

```
$ mkdir -p /apache/httpd
```

```
$ mkdir -p /apache/www
```

Tras la creación de estos dos directorios, se procederá a copiar en ellos, todo el contenido de los ficheros de configuración y de contenido web de Apache y de Wordpress (ya que los archivos de *Wordpress* están bajo el directorio de contenidos de Apache). Para ello, desde el Nodo1 se ejecutará:

```
$ cp -r /var/www/* /apache/www/
```

```
$ cp -r /etc/httpd/* /apache/httpd/
```

Una vez copiado todo el contenido de los directorios de configuración y de contenido web, de Apache y Wordpress, se deben ajustar determinados contextos de seguridad de SELinux para el correcto funcionamiento de Apache y Wordpress desde el Nodo1 (Ver ***Ilustración 143***):

```
[root@nodo1 ~]# cd /apache/
[root@nodo1 apache]# runcon -u system_u /bin/bash
[root@nodo1 apache]# semanage fcontext -a -t httpd_config_t "/apache/httpd"
[root@nodo1 apache]# semanage fcontext -a -t httpd_config_t "/apache/httpd(/.*)?"
[root@nodo1 apache]# semanage fcontext -a -t httpd_sys_content_t "/apache/www/"
[root@nodo1 apache]# semanage fcontext -a -t httpd_sys_content_t "/apache/www/html"
[root@nodo1 apache]# semanage fcontext -a -t httpd_sys_script_exec_t "/apache/www/cgi-bin"
[root@nodo1 apache]# semanage fcontext -a -t httpd_sys_content_t "/apache/www/html(/.*)?"
[root@nodo1 apache]# semanage fcontext -a -t httpd_sys_script_exec_t "/apache/www/cgi-bin(/.*)?"
[root@nodo1 apache]# _
```

Ilustración 143: Contextos de SELinux de los ficheros de Apache y de Wordpress en los Nodos 1 y 2

Una vez modificados los contextos de SELinux, se debe ejecutar desde el Nodo1, el siguiente comando para que se apliquen adecuadamente:

```
$ restorecon -R -v /apache/
```

Y se deben cambiar el usuario y grupo propietario de los ficheros bajo el directorio ***"/apache/www/html"***:

```
$ chown -R /apache:apache /apache/www/html/
```

Una vez en este punto, es recomendable reiniciar (*reboot*) tanto el nodo1 como el Nodo2.

Tras esto, se deben renombrar los directorios ***"/etc/httpd/conf"***, ***"/etc/httpd/conf.d"*** y ***"/var/www"*** a modo de copia de seguridad haciendo uso de comando "*mv*" como se muestra en la siguiente ilustración en el Nodo1 y 2:

```
[root@nodo1 /]# cd /etc/httpd/
[root@nodo1 httpd]# ls
conf  conf.d  conf.modules.d  logs  modules  run  state
[root@nodo1 httpd]# mv conf conf_copia
[root@nodo1 httpd]# mv conf.d conf.d_copia
[root@nodo1 httpd]# cd /var
[root@nodo1 var]# ls
adm    crash  empty  games   kerberos  local  log   nis  preserve  spool  www
cache  db     ftp    gopher  lib       lock   mail  opt  run       tmp    yp
[root@nodo1 var]# mv www www_copia
[root@nodo1 var]# _
```

Ilustración 144: Copias de seguridad de los directorios de configuración de Apache

Por último, se deben crear los enlaces simbólicos de manera que apunten desde los directorios de la ruta original de los ficheros de configuración de Apache, hacia el directorio del espacio compartido (**/apache**). Para ello, se deben crear los enlaces simbólicos en el nodo 1 y en el Nodo 2, tal y como se muestran en la siguiente ilustración:

```
[root@nodo1 httpd]# ln -s /apache/httpd/conf /etc/httpd/conf
[root@nodo1 httpd]# ln -s /apache/httpd/conf.d /etc/httpd/conf.d
[root@nodo1 httpd]# ln -s /apache/www /var/www
[root@nodo1 httpd]# _
```

Ilustración 145: Enlace simbólicos a los ficheros de configuración de Apache

Con los enlaces simbólicos creados, se tiene un espacio compartido por los Nodos 1 y 2 bajo el directorio "**/apache**", para el almacenamiento de los ficheros de configuración del servidor web *Apache,* con acceso concurrente síncrono y alta disponibilidad.

Anexo XXXIV: Pasos para lograr acceso concurrente y alta disponibilidad en los ficheros de configuración de *MariaDB*

En primer lugar, se creará un volumen lógico de 2GB llamado "**lv_mariadb_server**" que haga uso del espacio disponible en el grupo de volúmenes "**vg_almacenamiento_compartido**". Desde el Nodo3, se creará este volumen lógico de la siguiente manera:

```
$ lvcreate -n lv_mariadb_server -L 2G vg_almacenamiento_compartido
```

Una vez creado el volumen lógico anterior en el que se guardarán los ficheros de configuración de MariaDB, se procederá a crear sobre él, un sistema de ficheros OCFS2 para otorgar acceso concurrente y alta disponibilidad a sus archivos.

```
$ mkfs.ocfs2 -N 2 /dev/mapper/vg_almacenamiento_compartido-
lv_mariadb_server
```

Nota: La opción "**-N**", indica el número de nodos que tienen acceso simultáneo al volumen.

Tras el paso anterior, es muy recomendable reiniciar todos los nodos del clúster OCFS2 (*reboot*), para que la configuración del volumen lógico recién creado se aplique en todos los nodos del clúster. Una vez arrancados los nodos, se debe verificar con la orden "*lsblk*" que el volumen lógico para MariaDB y su formato, aparecen correctamente en todos los nodos, ya que este volumen lógico está establecido sobre el almacenamiento compartido iSCSI.

Lo siguiente que se realizará, será la creación del directorio "**/mariadb**" en los Nodos 3 y 4, que albergará todos los ficheros de configuración de MariaDB.

Para ello:

```
$ mkdir -p /mariadb
```

Una vez creado el directorio para MariaDB en los nodos 3 y 4, se debe montar el volumen lógico correspondiente a MariaDB tanto en el Nodo3 como en el Nodo4. Además, se debe montar de tal manera que se monte automáticamente en el arranque de ambos nodos. Para ello, se debe modificar el fichero ***"/etc/fstab***" de los Nodos 3 y 4, añadiendo al final del mismo la siguiente línea:

```
/dev/mapper/vg_almacenamiento_compartido-lv_mariadb_server /mariadb ocfs2 _netdev 0 0
```

Nota: Se usa la opción "***_netdev***", porque el sistema de archivos reside en un dispositivo que requiere acceso a la red.

Tras modificar el fichero "**fstab**" de los Nodos 3 y 4, se debe ejecutar la orden "*mount -a*" en los Nodos 3 y 4, para que se monte todo lo que se ha establecido en el "fstab".

Tras realizar todos los pasos anteriores, se ha montado el volumen lógico para los ficheros de configuración de *MariaDB*, en el directorio "**/mariadb**" de los nodos 3 y 4. Por lo tanto, llegados a este punto, se tiene acceso concurrente y síncrono en el directorio "/mariadb" de los Nodos 3 y 4, proporcionando de esta manera alta disponibilidad en el almacenamiento compartido.

Lo siguiente que se debe hacer, es crear el directorio "**my.cnf.d**" correspondiente al servicio MariaDB en el directorio "/mariadb" de los Nodos 3 y 4. Para la creación de este directorio, tan solo hará falta crearlo en uno de los dos nodos (Nodo3 o Nodo4), ya que ahora, el directorio "/mariadb" dispone de acceso concurrente y síncrono, garantizando que los datos creados en este

directorio, en uno de los nodos, se replican automáticamente en el otro nodo (alta disponibilidad). Para ello, se procederá creando este directorio desde el Nodo3:

```
$ mkdir -p /mariadb/my.cnf.d
```

Tras la creación de este directorio, se procederá a copiar en él y en el directorio "*/mariadb*", todo el contenido de los ficheros de configuración de MariaDB. Para ello, desde el Nodo3 se ejecutará:

```
$ cp -r /etc/my.cnf /mariadb
```

```
$ rsync -a --exclude 'gale*' /etc/my.cnf.d/ /mariadb/my.cnf.d/
```

Nota: Para copiar todos los ficheros de configuración de MariaDB que están bajo el directorio ***"/etc/my.cnf.d"***, se ha hecho uso de la utilidad "**rsync**" y la opción **"--exclude**", para copiar en el directorio ***"/mariadb/my.cnf.d"*** todos sus ficheros excepto los ficheros de Galera, ya que estos ficheros son independientes en cada nodo.

Una vez copiado todo el contenido de los directorios de configuración de MariaDB, se deben ajustar determinados contextos de seguridad de SELinux para el correcto funcionamiento de MariaDB desde el Nodo3 (Ver ***Ilustración 146***):

```
[root@nodo3 ~]# cd /mariadb/
[root@nodo3 mariadb]# runcon -u system_u /bin/bash
[root@nodo3 mariadb]# semanage fcontext -a -t mysqld_etc_t "/mariadb/my.cnf"
[root@nodo3 mariadb]# semanage fcontext -a -t mysqld_etc_t "/mariadb/my.cnf.d"
[root@nodo3 mariadb]# semanage fcontext -a -t mysqld_etc_t "/mariadb/my.cnf.d(/.*)?"
```

Ilustración 146: Contextos de SELinux de los ficheros de configuración de MariaDB en los Nodos 3 y 4

Una vez modificados los contextos de SELinux, se debe ejecutar desde el Nodo3, el siguiente comando para que se apliquen adecuadamente:

```
$ restorecon -R -v /mariadb/
```

Una vez en este punto, es recomendable reiniciar (*reboot*) tanto el nodo3 como el Nodo4.

Tras esto, se deben renombrar los archivos ***"/etc/my.cnf"***, ***"client.cnf"***, ***"enable_encryption.preset", "mariadb-server.cnf"*** *y* ***"mysql-clients.cnf",*** a modo de copia de seguridad haciendo uso de comando "*mv*" como se muestra en la siguiente ilustración en los nodos 3 y 4:

Ilustración 147: Copias de seguridad de los directorios de configuración de MariaDB

Por último, se deben crear los enlaces simbólicos de manera que apunten desde los directorios de la ruta original de los ficheros de configuración de MariaDB, hacia el directorio del espacio compartido (**/mariadb**). Para ello, se deben crear los enlaces simbólicos en el nodo 3 y en el Nodo 4, tal y como se muestran en la siguiente ilustración:

Ilustración 148: Enlace simbólicos a los ficheros de configuración de MariaDB

Tras esto, se deberá volver a iniciar el clúster de Galera (***galera_new_cluster***) y agregar el segundo nodo al mismo.

Con los enlaces simbólicos creados, se tiene un espacio compartido por los Nodos 3 y 4 bajo el directorio "**/mariadb**", para el almacenamiento de los ficheros de configuración del servicio MariaDB, con acceso concurrente síncrono y alta disponibilidad.

19. Fuentes de Información

[1] "Sistemas de alta disponibilidad: Definición, importancia y principios", Historiadelaempresa.com, 2022 [En línea]. Disponible en: https://onx.la/8b772 . [Accedido: 24/05/2022]

[2] J. Jiménez, "Qué es Alta Disponibilidad o HA y por qué es importante en servidores", Redes Zone, 2020 [En línea]. Disponible en: https://www.redeszone.net/tutoriales/servidores/alta-disponibilidad-importante-servidores/ . [Accedido: 24/05/2022]

[3] H. Wacker, "¿Qué es el balanceo de carga?", COMPUTERWORLD, 2000 [En línea]. Disponible en: https://www.computerworld.es/tendencias/que-es-el-balanceo-de-carga .[Accedido: 24/05/2022]

[4] "Elige un cluster para asegurar la disponibilidad de tu proyecto", Linube, 2020 [En línea]. Disponible en: https://linube.com/blog/cluster-web-que-es/ . [Accedido: 24/05/2022]

[5] E. Medina, "CentOS 8 será descontinuado en 2021 para convertirse en «rolling release»", MUY LINUX, 2020 [En línea]. Disponible en: https://onx.la/30d9d. [Accedido: 24/05/2022]

[6] “Objetivos y competencias del GII”, ULPGC, 2022 [En línea]. Disponible en: https://www2.ulpgc.es/archivos/plan_estudios/4008_40/ObjetivosyCompetenciasdelGII.pdf . [Accedido: 24/05/2022]

[7] “Licencias de software”, tic.PORTAL, 2022 [En línea]. Disponible en: https://www.ticportal.es/glosario-tic/licencias-software . [Accedido: 05/07/2022]

[8] “GNU General Public License”, Wikipedia, 2022 [En línea]. Disponible en: https://es.wikipedia.org/wiki/GNU_General_Public_License. [Accedido: 05/07/2022]

[9] “Apache License”, Wikipedia, 2022 [En línea]. Disponible en: https://es.wikipedia.org/wiki/Apache_License#Versi%C3%B3n_2.0 .[Accedido: 05/07/2022]

[10] “Licencia de PHP”, Wikipedia, 2022 [En línea]. Disponible en: https://es.wikipedia.org/wiki/Licencia_PHP .[Accedido: 05/07/2022]

[11] E. Limones, “Virtualización: Qué es, para qué sirve y ventajas”, OpenWebinars.net, 2021 [En línea]. Disponible en: https://openwebinars.net/blog/virtualizacion-que-es-para-que-sirve-y-ventajas/ . [Accedido: 05/07/2022]

[12] F. A. Quesada Arencibia y C. R. García Rodríguez, “Tema 1: Fundamentos y Tecnologías de Virtualización”, Infraestructuras Tecnológicas para los Sistemas de Información, 2021. [En línea]. Disponible en: https://aep22.ulpgc.es/pluginfile.php/786493/mod_resource/content/8/Tema1_Introduccion.pdf . [Accedido: 05/07/2022]

[13] “Virtualización”, Wikipedia, 2022 [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Virtualizaci%C3%B3n&oldid=143042652 . [Accedido: 05/07/2022]

[14] “¿Qué es un cluster y para qué sirve?”, Blog de Zendesk, 2021. [En línea]. Disponible en: https://www.zendesk.com.mx/blog/cluster-que-es/ . [Accedido: 05/07/2022]

[15] F. A. Quesada Arencibia y C. R. García Rodríguez, “Computación en Clúster”, Infraestructuras Tecnológicas para los Sistemas de Información, 2021. [En línea]. Disponible en: https://aep22.ulpgc.es/pluginfile.php/786542/mod_resource/content/3/COMPUTACION%20EN%20%20CLUSTER.pdf . [Accedido: 06/07/2022]

[16] “Oracle Linux”, Oracle.com, 2022. [En línea]. Disponible en: https://www.oracle.com/es/linux/ . [Accedido: 06/07/2022]

[17] “Kernel Virtual Machine”, KVM, 2022. [En línea]. Disponible en: https://www.linux-kvm.org/page/Main_Page . [Accedido: 06/07/2022]

[18] “Managing Software in Oracle Linux - DNF Command Reference”, Oracle.com, 2022. [En línea]. Disponible en: https://docs.oracle.com/en/operating-systems/oracle-linux/software-management/sfw-mgmt-DNFCommandReference.html#dnf-command-ref . [Accedido: 06/07/2022]

[19] “What is MariaDB Galera Cluster?”, MariaDB.com, 2021. [En línea]. Disponible en: https://mariadb.com/kb/en/what-is-mariadb-galera-cluster/ . [Accedido: 06/07/2022]

[20] “HAProxy - The Reliable, High Performance TCP/HTTP Load Balancer”, HAProxy.org, 2022. [En línea]. Disponible en: https://www.haproxy.org/ . [Accedido: 06/07/2022]

[21] “Install the Apache Web Server - Introduction”, Oracle.com, 2022. [En línea]. Disponible en: https://docs.oracle.com/en/learn/apache-install/index.html#introduction . [Accedido: 06/07/2022]

[22] “Despliegue de diferentes tipos de servidores – Introducción a MariaDB”, Red Hat 8.0, 2022. [En línea]. Disponible en: https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/8/html/deploying_different_types_of_servers/using-mariadb#introduction-to-mariadb_using-mariadb . [Accedido: 06/07/2022]

[23] “¿Qué es PHP?”, PHP.net, 2022. [En línea]. Disponible en: https://www.php.net/manual/es/intro-whatis.php . [Accedido: 06/07/2022]

[24] “¿Qué es WordPress, Para Qué Sirve y Cómo Funciona?”, InstitutoCajasol.com, 2022. [En línea]. Disponible en: https://institutocajasol.com/que-es-wordpress-y-como-funciona/. [Accedido: 06/07/2022]

[25] Margaret, “¿Qué es el almacenamiento iSCSI y cómo crear una SAN iSCSI?”, Community.fs, 2021. [En línea]. Disponible en: https://community.fs.com/es/blog/iscsi-storage-basics-plan-iscsi-san.html . [Accedido: 06/07/2022]

[26] “Uso de SELinux - Introducción a SELinux”, Red Hat 8.0, 2021. [En línea]. Disponible en: https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/8/html-single/using_selinux/index#introduction-to-selinux_getting-started-with-selinux . [Accedido: 06/07/2022]

[27] “Configuración y gestión de redes - FirewallD”, Red Hat 8.0, 2021. [En línea]. Disponible en: https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/8/html/configuring_and_managing_networking/getting-started-with-firewalld_using-and-configuring-firewalld#firewalld_getting-started-with-firewalld . [Accedido: 06/07/2022]

[28] “OCFS2”, Wikipedia, 2021. [En línea]. Disponible en: https://es.wikipedia.org/wiki/OCFS2 . [Accedido: 06/07/2022]

[29] “Oracle Linux Installation Media”, Oracle.com, 2022. [En línea]. Disponible en: https://yum.oracle.com/oracle-linux-isos.html . [Accedido: 06/07/2022]

[30] “Flash OS images to SD cards & USB drives, safely and easily.”, Balena.io, 2022. [En línea]. Disponible en: https://www.balena.io/etcher/ . [Accedido: 06/07/2022]

[31] T. Carrigan, “How to change your hostname in Linux”, Red Hat, 2020. [En línea]. Disponible en: https://www.redhat.com/sysadmin/change-hostname-linux. [Accedido: 07/07/2022]

[32] “KVM User's Guide”, Oracle.com, 2022. [En línea]. Disponible en: https://docs.oracle.com/en/operating-systems/oracle-linux/kvm-user/ . [Accedido: 07/07/2022]

[33] “Cómo administrar máquinas virtuales KVM con Virt-Manager”, MuyLinux.xyz, 2021. [En línea]. Disponible en: https://muylinux.xyz/como-administrar-maquinas-virtuales-kvm-con-virt-manager/ . [Accedido: 08/07/2022]

[34] F. A. Quesada Arencibia y C. R. García Rodríguez, “Operaciones con MVs”, Infraestructuras Tecnológicas para los Sistemas de Información, 2021. [En línea]. Disponible en: https://aep22.ulpgc.es/pluginfile.php/786640/mod_resource/content/3/P3_Operaciones_MV.pdf . [Accedido: 08/07/2022]

[35] “Configuración y gestión de la virtualización”, Red Hat 8.0, 2022. [En línea]. Disponible en: https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/8/html-single/configuring_and_managing_virtualization/index#doc-wrapper . [Accedido: 08/07/2022]

[36] “Random MAC Address Generator”, Mac Address Lookup, 2022. [En línea]. Disponible en: https://www.macvendorlookup.com/mac-address-generator . [Accedido: 08/07/2022]

[37] F. A. Quesada Arencibia y C. R. García Rodríguez, “Almacenamiento en KVM”, Infraestructuras Tecnológicas para los Sistemas de Información, 2021. [En línea]. Disponible en: https://aep22.ulpgc.es/pluginfile.php/786703/mod_resource/content/1/P4_RecursosAlmacenamientoVirtual_KVM.pdf . [Accedido: 08/07/2022]

[38] J. Herrmann, P. Shah, Y. Zimmerman, L. Novich, D. Parker, S. Radvan, T. Richardson “Virtualization Deployment and Administration Guide”, Red Hat 7.0, 2021. [En línea]. Disponible en: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/virtualization_deployment_and_administration_guide/index . [Accedido: 08/07/2022]

[39] “Managing file systems”, Red Hat 8.0, 2022. [En línea]. Disponible en: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/managing_file_systems/index . [Accedido: 08/07/2022]

[40] F. A. Quesada Arencibia y C. R. García Rodríguez, “Práctica 5: Recursos de almacenamiento virtual.”, Infraestructuras Tecnológicas para los Sistemas de Información, 2021. [En línea]. Disponible en: https://aep22.ulpgc.es/pluginfile.php/786696/mod_resource/content/16/Ficha_P5_Recursos_Almacenamiento_Virtual.pdf . [Accedido: 08/07/2022]

[41] F. A. Quesada Arencibia y C. R. García Rodríguez, “Infraestructura de Red virtual”, Infraestructuras Tecnológicas para los Sistemas de Información, 2021. [En línea]. Disponible en: https://aep22.ulpgc.es/pluginfile.php/786717/mod_resource/content/5/P6_Infraestructura_Red_Virtual.pdf . [Accedido: 09/07/2022]

[42] “Chapter 5. Getting started with nmcli”, Red Hat 8.0, 2022. [En línea]. Disponible en: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/configuring_and_managing_networking/getting-started-with-nmcli_configuring-and-managing-networking. [Accedido: 10/07/2022]

[43] “Install the Apache Web Server”, Oracle.com, 2022. [En línea]. Disponible en: https://docs.oracle.com/en/learn/apache-install/index.html#installing-and-configuring-the-web-server-package .[Accedido: 11/07/2022]

[44] “Setting Up Load Balancing by Using HAProxy”, Oracle.com, 2022. [En línea]. Disponible en: https://docs.oracle.com/en/operating-systems/oracle-linux/8/balancing/balancing-SettingUpLoadBalancingbyUsingHAProxy.html#haproxy-config .[Accedido: 11/07/2022]

[45] S. Biradar, “Configure Apache HAProxy to balance web server traffic”, Red Hat SysAdmin, 2022. [En línea]. Disponible en: https://www.redhat.com/sysadmin/configure-apache-haproxy-load-balance .[Accedido: 11/07/2022]

[46] “Setting Up High Availability Clustering”, Oracle.com, 2022. [En línea]. Disponible en: https://docs.oracle.com/en/operating-systems/oracle-linux/8/availability/index.html .[Accedido: 11/07/2022]

[47] “What is fencing | Setup KVM cluster fencing RHEL CentOS 8”, GoLinuxCloud.com, 2020. [En línea]. Disponible en: https://www.golinuxcloud.com/what-is-fencing-configure-kvm-cluster-fencing/ .[Accedido: 11/07/2022]

[48] F. A. Quesada Arencibia y C. R. García Rodríguez, “Práctica 7.2: Diseño y despliegue de un clúster básico”, Infraestructuras Tecnológicas para los Sistemas de Información, 2021. [En línea]. Disponible en: https://aep22.ulpgc.es/pluginfile.php/786759/mod_resource/content/15/Ficha_P7_2_Cluster_Basico.pdf . [Accedido: 11/07/2022]

[49] “Chapter11. Configuring location constraints”, Red Hat 8.0, 2022. [En línea]. Disponible en: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/configuring_and_managing_high_availability_clusters/assembly_determining-which-node-a-resource-runs-on-configuring-and-managing-high-availability-clusters .[Accedido: 11/07/2022]

[50] “Despliegue del clúster MariaDB Galera”, Red Hat 8.0, 2022. [En línea]. Disponible en: https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/8/html/deploying_different_types_of_servers/configuring-mariadb-galera-cluster_replicating-mariadb-with-galera .[Accedido: 11/07/2022]

[51] E. Moreira, “Como Crear y Configurar un Cluster de Base de Datos de MariaDB con Galera en CentOS 8 / RHEL 8 / AlmaLinux 8 / Rocky Linux 8”, It’s Simple Now, 2022. [En línea]. Disponible en: https://www.itsimplenow.com/como-configurar-un-cluster-de-mariadb-con-galera-en-centos-8-rhel-8/#promo-digitalocean .[Accedido: 11/07/2022]

[52] “HAProxy”, GaleraCluster.com, 2022. [En línea]. Disponible en: https://galeracluster.com/library/documentation/ha-proxy.html .[Accedido: 11/07/2022]

[53] Amalfitano, “Estadísticas en HAProxy”, l10nn.medium, 2018. [En línea]. Disponible en: https://l10nn.medium.com/estad%C3%ADsticas-en-haproxy-397a2183844 .[Accedido: 12/07/2022]

[54] V. Dusa, “Cómo instalar el stack LAMP en Oracle Linux”. OurcodeWordl, 2021. [En línea]. Disponible en: https://ourcodeworld.co/articulos/leer/1445/como-instalar-el-stack-lamp-en-oracle-linux .[Accedido: 12/07/2022]

[55] “Install WordPress CMS”. Oracle.com, 2022. [En línea]. Disponible en: https://docs.oracle.com/en/learn/wrdprs_mysqldbs_wrkshp/index.html#install-wordpress-cms .[Accedido: 12/07/2022]

[56] “How To Setup High Availability Cluster on CentOS 8 / RHEL 8”. IT’zGeek, 2020. [En línea]. Disponible en: https://www.itzgeek.com/post/how-to-setup-high-availability-cluster-on-centos-8-rhel-8/ .[Accedido: 12/07/2022]

[57] “Use Oracle Cloud Cluster File System Tools on Oracle Cloud Infrastructure”. Oracle.com, 2022. [En línea]. Disponible en: https://docs.oracle.com/en/learn/ocfs2_cluster_linux_8/index.html#configure-the-cluster-layout .[Accedido: 12/07/2022]

[58] “Managing Shared File Systems”. Oracle.com, 2022. [En línea]. Disponible en: https://docs.oracle.com/en/operating-systems/oracle-linux/8/shareadmin/shareadmin-ManagingtheOracleClusterFileSystemVersion2inOracleLinux.html#ocfs2 .[Accedido: 12/07/2022]