

Implantación de un sistema de Comercio Electrónico virtualizado en configuración distribuida y tolerante a fallos.

José Ulla Carrera

Grado en Ingeniería Informática

Administración de Redes y Sistemas Operativos

Mario Prieto Vega

David Bañeres Besora

Montse Serra Vizern

14/01/2023



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Implantación de un sistema de Comercio Electrónico virtualizado en configuración distribuida y tolerante a fallos.</i>
Nombre del autor:	<i>José Ulla Carrera</i>
Nombre del consultor/a:	<i>Mario Prieto Vega</i>
Nombre del PRA:	<i>Nombre y dos apellidos</i>
Fecha de entrega (mm/aaaa):	01/2023
Titulación:	<i>Grado en Ingeniería Informática</i>
Área del Trabajo Final:	<i>Administración de Redes y Sistemas Operativos</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Virtualización E-Commerce Tolerancia a fallos</i>

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

La motivación de este proyecto nace de la necesidad de adaptar los *E-Commerce* modernos, de pequeñas y medianas empresas, a la gran variabilidad de las ventas y concurrencia de usuarios que existen debido a las campañas de marketing y periodos de ofertas, tales como Black Friday, campaña de navidad, San Valentín etc.

El sistema monolítico clásico, con un único servidor, físico o virtual, sólo escalable verticalmente, que ejecuta exclusivamente todos los componentes y servicios software necesarios para poder ejecutar un *E-Commerce*, tiene serias carencias para adaptarse a estas variabilidades tan grandes de tráfico y concurrencia de usuarios.

Para lograr la plasticidad necesaria a estos sistemas para adaptarse a distintas demandas de tráfico, se propone una arquitectura basada en Software Libre con un sistema de virtualización basado en un hipervisor Proxmox, en versión 7 sobre distribución Debian 11, sobre el que se ejecutarán máquinas virtuales o contenedores usando también Debian 11 y Debian 10, los cuales estarán separados en servicios distribuidos, codependientes del *E-Commerce*, y a su vez replicados de forma que el sistema sea fácilmente escalable, tanto a nivel de recursos como a nivel de entorno, es decir que sea fácilmente replicable en otro hipervisor preexistente, nube pública etc.

Abstract (in English, 250 words or less):

The motivation for this project stems from the need to adapt modern *E-Commerce* of small and medium sized businesses to the great variability of sales and user concurrence that exist due to marketing campaigns and offer periods, such as Black Friday, Christmas campaign, San Valentine, etc.

The classic monolithic system, with a single server, physical or virtual, only vertically scalable, which exclusively executes all the software components and services necessary to be able to run an E-Commerce, has serious shortcomings to adapt to these large variations in traffic and concurrence. of users.

To achieve the plasticity necessary for these systems to adapt to different traffic demands, an architecture based on Free Software is proposed with a virtualization system based on a Proxmox Hypervisor, in version 7 on Debian 11 distribution, on which virtual machines or containers will be executed, also using Debian 11 and Debian 10, which will be separated into distributed services, co-dependent on E-Commerce, and at the same time replicated so that the system is easily scalable, both at the resource level and at the environment level, that is to say, that it is easily replicable in another pre-existing hypervisor, public cloud, etc.

Índice

1. Introducción.....	1
1.1. Contexto y justificación del Trabajo.....	1
1.2. Objetivos del Trabajo.....	2
1.3. Enfoque y método seguido.....	4
1.4. Planificación del Trabajo.....	4
Hito 1. Propuesta de plan de trabajo.....	5
Hito 2. Entrega entre el 40% y el 60% del total del TFG.....	5
Hito 3. Entrega entre el 80% y 90% del total de la memoria.....	6
Hito 4. Entrega final.....	6
1.5. Breve resumen de productos obtenidos.....	7
1.6. Breve descripción de los otros capítulos de la memoria.....	7
2. Marco del trabajo.....	8
2.1. Comercio electrónico.....	8
2.2. Sistemas monolíticos.....	9
2.3. Sistemas distribuidos.....	10
2.4. Clústeres y Microservicios.....	11
2.5. Virtualización y contenedores.....	12
3. Arquitectura.....	13
3.1. Hardware e Hipervisor.....	13
3.2. Configuración de Red.....	13
3.3. Configuración de Almacenamiento.....	14
3.4. Configuración de Copias de Seguridad.....	15
3.5. Máquinas de Infraestructura.....	16
4. Software.....	17
4.1. Sistemas Operativos.....	17
4.2. GlusterFS.....	18
4.3. PHP.....	22
4.4. MySQL/MariaDB.....	25
4.5. Redis.....	28
4.6. Elasticsearch.....	31
4.7. Varnish y Nginx.....	32
4.8. Adobe Commerce Magento <i>Open source</i>	35
5. Alta disponibilidad y tolerancia a fallos.....	40
5.1. Replicación de rutas.....	41
5.2. Redundancia de balanceo y caché de página.....	43
5.3. Redundancia de intérpretes de PHP.....	44
5.4. Clúster de base de datos relacional MariaDB.....	45
5.5. Clúster de base de datos de objetos Redis.....	49
5.6. Clúster de servicio de búsqueda Elasticsearch.....	50
5.7. Ilustración final de la infraestructura.....	53
6. Conclusiones.....	55
6.1. Puntos de Mejora.....	56
7. Glosario.....	58
8. Bibliografía.....	63
9. Anexos.....	65

9.1 Sistema de copias de seguridad.	65
9.2. Listado de IP:	67
9.3. Archivos de configuración modificados para GlusterFS:	67
9.4. Archivos de configuración modificados para PHP.....	68
9.5. Archivos de configuración modificados para MariaDB.	69
9.6. Archivos de configuración modificados para Redis.	71
9.7. Archivos de configuración modificados para Elasticsearch.....	73
9.7. Archivos de configuración modificados para Varnish.	74
9.9. Archivos de configuración modificados y creados para Nginx.....	78
9.10. Archivos de configuración modificados para Magento	81

Lista de figuras

Ilustración 1 - Planificación de TFG.	5
Ilustración 2 - Hito 1. Propuesta de plan de trabajo.	5
Ilustración 3 - Hito 2. Entrega entre el 40% y el 60% del total del TFG	6
Ilustración 4 - Hito 3. Entrega entre el 80% y 90% del total de la memoria.	6
Ilustración 5 - Hito 4. Entrega final.	7
Ilustración 6 - Ejemplo gráfico de sistema monolítico.	9
Ilustración 7 - Ejemplo gráfico de sistema distribuido.	10
Ilustración 8 - Ejemplo gráfico de infraestructura virtual.	12
Ilustración 9 - Proxmox VE (Virtual Environment).	13
Ilustración 10 - Proxmox VE. Configuración de Red.	14
Ilustración 11 - Proxmox VE. Programación copias de seguridad.	15
Ilustración 12 - Proxmox Backup Server. Datastore UOCTFG.	16
Ilustración 13 - Proxmox VE. Pool de máquinas de infraestructura.	17
Ilustración 14 - Proxmox VE. Pool de máquinas de almacenamiento.	18
Ilustración 15 - Instalación de GlusterFS 1.	19
Ilustración 16 - Instalación de GlusterFS 2.	19
Ilustración 17 - Instalación de GlusterFS 3.	20
Ilustración 18 - Instalación de GlusterFS 4.	20
Ilustración 19 - Instalación de GlusterFS 5.	20
Ilustración 20 - Instalación de GlusterFS 6.	21
Ilustración 21 - Instalación de GlusterFS 7.	21
Ilustración 22 - Instalación de GlusterFS 8.	21
Ilustración 23 - Instalación de GlusterFS 9.	21
Ilustración 24 - Instalación de GlusterFS 10.	21
Ilustración 25 - Instalación de GlusterFS 11.	22
Ilustración 26 - Instalación de GlusterFS 12.	22
Ilustración 27 - Instalación de GlusterFS 13.	22
Ilustración 28 - Instalación de PHP 1.	23
Ilustración 29 - Instalación de PHP 2.	23
Ilustración 30 - Instalación de PHP 3.	24
Ilustración 31 - Instalación de PHP 4.	24
Ilustración 32 - Instalación de PHP 5.	24
Ilustración 33 - Instalación de MariaDB 1.	26
Ilustración 34 - Instalación de MariaDB 2.	26
Ilustración 35 - Instalación de MariaDB 3.	26
Ilustración 36 - Instalación de MariaDB 4.	26
Ilustración 37 - Instalación de MariaDB 5.	27
Ilustración 38 - Instalación de MariaDB 6.	28
Ilustración 39 - Instalación de MariaDB 7.	28
Ilustración 40 - Instalación de MariaDB 8.	28
Ilustración 41 - Instalación de MariaDB 9.	28
Ilustración 42 - Instalación de Redis 1.	29
Ilustración 43 - Instalación de Redis 2.	29
Ilustración 44 - Instalación de Redis 3.	29
Ilustración 45 - Instalación de Redis 4.	30
Ilustración 46 - Instalación de Redis 5.	30

Ilustración 47 - Instalación de Redis 6.	30
Ilustración 48 - Instalación de Redis 7.	30
Ilustración 49 - Instalación de Redis 8.	30
Ilustración 50 - Instalación de Elasticsearch 1.	31
Ilustración 51 - Instalación de Elasticsearch 2.	31
Ilustración 52 - Instalación de Elasticsearch 3.	31
Ilustración 53 - Instalación de Elasticsearch 4.	32
Ilustración 54 - Instalación de Elasticsearch 5.	32
Ilustración 55 - Instalación de Elasticsearch 6.	32
Ilustración 56 - Instalación de Elasticsearch 7.	32
Ilustración 57 - Instalación de Varnish 1.	33
Ilustración 58 - Instalación de Varnish 2.	33
Ilustración 59 - Instalación de Nginx 1.	34
Ilustración 60 - Instalación de Nginx 2.	34
Ilustración 61 - Flujo de entrada de peticiones de cliente.	35
Ilustración 62 - Instalación de Adobe Commerce Magento <i>Open source</i> 1.	36
Ilustración 63 - Instalación de Adobe Commerce Magento <i>Open source</i> 2.	36
Ilustración 64 - Instalación de Adobe Commerce Magento <i>Open source</i> 3.	37
Ilustración 65 - Instalación de Adobe Commerce Magento <i>Open source</i> 4.	37
Ilustración 66 - Instalación de Adobe Commerce Magento <i>Open source</i> 5.	37
Ilustración 67 - Instalación de Adobe Commerce Magento <i>Open source</i> 6.	37
Ilustración 68 - Instalación de Adobe Commerce Magento <i>Open source</i> 7.	37
Ilustración 69 - Instalación de Adobe Commerce Magento <i>Open source</i> 8.	37
Ilustración 70 - Adobe Commerce Magento <i>Open source</i> . Login Backoffice.	38
Ilustración 71 - Adobe Commerce Magento <i>Open source</i> . Configuración full page caché 1.	38
Ilustración 72 - Adobe Commerce Magento <i>Open source</i> . Configuración de full page caché 2.	39
Ilustración 73 - Adobe Commerce Magento <i>Open source</i> . Configuración de full page caché 3.	39
Ilustración 74 - Adobe Commerce Magento <i>Open source</i> . Configuración Store Search Engine.	40
Ilustración 75 - Adobe Commerce Magento <i>Open source</i> . <i>Frontend</i> .	40
Ilustración 76 - Configuración de red de VM Router.	42
Ilustración 77 - Redundancia de rutas.	42
Ilustración 78 - Configuración NAT routing final.	43
Ilustración 79 - Configuración balanceadores final.	43
Ilustración 80 - Configuración de caché de página final 1.	44
Ilustración 81 - Configuración de caché de página final 2.	44
Ilustración 82 - Configuración intérpretes PHP final.	45
Ilustración 83 - Configuración balanceo final.	45
Ilustración 84 - Replicación MariaDB - Configuración servidor activo.	46
Ilustración 85 - Replicación MariaDB - Configuración permisos.	46
Ilustración 86 - Replicación MariaDB - Estado inicial servidor activo.	46
Ilustración 87 - Replicación MariaDB - Configuración servidor pasivo.	47
Ilustración 88 - Replicación MariaDB - Parada de la replicación.	47
Ilustración 89 - Replicación MariaDB - Configuración y arranque de replicación.	47
Ilustración 90 - Replicación MariaDB - Estado inicial.	48

Ilustración 91 - Réplicación MariaDB - Estado de logs tras un cambio en Backoffice.	48
Ilustración 92 - Réplicación MariaDB - Réplicación exitosa.	49
Ilustración 93 - Réplicación Redis - Configuración servidor secundario.	50
Ilustración 94 - Réplicación Redis - Estado primario.	50
Ilustración 95 - Réplicación Redis - Estado secundario.	50
Ilustración 96 - Clúster Elasticsearch - Configuración nodo els1.	51
Ilustración 97 - Clúster Elasticsearch - Configuración nodo els2.	51
Ilustración 98 - Clúster Elasticsearch - Configuración nodo els3.	51
Ilustración 99 - Cluster Elasticsearch - Modificación parámetros JVM.	52
Ilustración 100 - Cluster Elasticsearch - Comprobación del clúster.	52
Ilustración 101 - Configuración DNS cluster Elasticsearch.	52
Ilustración 102 - Configuración final Elasticsearch Magento.	53
Ilustración 103 - Arquitectura final.	54
Ilustración 104 - Datastore copias de seguridad.	65
Ilustración 105 - Configuración almacenamiento copias de seguridad.	66
Ilustración 106 - Programación de copias de seguridad.	66

1. Introducción

1.1. Contexto y justificación del Trabajo

En las últimas décadas se ha vivido un crecimiento exponencial en el comercio electrónico, sobre todo gracias a la venta online, la cual prácticamente se ha convertido en la definición del término.

Debido a este inmenso crecimiento han aparecido nuevos actores en el modelo de venta online, las grandes superficies tradicionales han abrazado el concepto con mayor o menor éxito y por último los pequeños y medianos empresarios están empezando finalmente a abrirse camino en este nuevo paradigma del negocio.

La motivación de este proyecto nace de la necesidad de adaptar las tiendas online o tiendas *E-Commerce* modernas, de pequeñas y medianas empresas, a la gran variabilidad de las ventas y concurrencia de usuarios que existen debido a las campañas de marketing y periodos de ofertas, tales como Black Friday, campaña de navidad, San Valentín etc., pero haciendo hincapié en las limitaciones tecnológicas y económicas de estas, tratando siempre de maximizar el rendimiento de cada euro invertido durante el proceso.

Gran parte de estas tiendas online están ancladas en el sistema monolítico clásico, con un único servidor, físico o virtual, sólo escalable verticalmente, que ejecuta exclusivamente todos los componentes y servicios software necesarios para poder ejecutar un *E-Commerce*, tiene serias carencias para adaptarse a estas variabilidades tan grandes de tráfico y concurrencia de usuarios. Si la cantidad de tráfico o usuarios concurrentes conectados a la plataforma sube de forma no controlada y el servidor no tiene capacidad para satisfacer todas las peticiones, acabará por fallar dejando sin servicio a todos los usuarios. La única forma posible de evitar esta casuística consiste en el escalamiento vertical, es decir, ampliaciones de recursos con paradas de servicio provocando indisponibilidad de la plataforma, algo que es especialmente crítico en sistemas físicos, tanto por coste como por la posibilidad de un fallo de hardware derivado de la intervención.

Este proyecto plantea una arquitectura virtual, distribuida y tolerante a fallos usando un hipervisor de virtualización Proxmox, plataforma *E-Commerce* Adobe Commerce Magento *Open source*, sobre servidor Web Nginx, con trabajadores agnósticos PHP-FPM y bases de datos MariaDB o MySQL. Mediante el uso de esta arquitectura se podrán añadir recursos de forma horizontal en la plataforma minimizando el número de paradas de servicio y maximizando la disponibilidad de la plataforma.

Se propone una arquitectura basada en Software Libre comenzando por un sistema de virtualización basado en un Hipervisor Proxmox, en versión 7 sobre distribución Debian 11, sobre el que se ejecutarán máquinas virtuales o contenedores usando también Debian 11, los cuales estarán separados en servicios distribuidos, codependientes del *E-Commerce*, y a su vez altamente disponibles de forma que el sistema sea fácilmente escalable, tanto a nivel de

recursos como a nivel de entorno, es decir que sea fácilmente replicable en otro hipervisor preexistente, nube pública etc.

El hipervisor se instalará sobre una computadora con hardware de propósito general, de aquí en adelante referido como “servidor”, concretamente con un procesador Intel Core i7 6700k, 32Gb de RAM y un disco duro SSD de 250Gb para el sistema operativo del hipervisor y otro disco SSD de 500Gb para guardar los discos de las máquinas virtuales y/o contenedores creados.

Se crearán instancias virtuales específicas para:

- Servidor Web y Offloader TCP/IP
- Intérprete PHP
- Sistema gestor de base de datos SQL
- Sistema gestor de base de datos no-SQL o clave-valor
- Motor de búsqueda
- Caché
- Almacenamiento distribuido

Las instancias no susceptibles de ser accedidas externamente estarán aisladas completamente del exterior, siendo únicamente posible acceder a través del servicio web correspondiente, cifrado con SSL. Debido a esta circunstancia se creará una máquina específica de gobierno, la cual tenga acceso a través de intérprete de comandos, mediante protocolo seguro y no interactivo (SSH mediante autenticación por par de claves), y únicamente a través de esta se podrán acceder a los respectivos intérpretes de comandos de las máquinas internas, tales como clientes de SGBD, consola de *E-Commerce* etc.

Se harán copias de seguridad diarias de todas las máquinas/contenedores, siempre y cuando el servidor se encuentre encendido durante las ventanas de copia establecidas. Este aspecto, a priori, no entra dentro del alcance del trabajo de fin de grado.

El software *E-Commerce* que se usará en el Trabajo de Fin de Grado es Adobe Commerce Magento *Open source* que tiene los siguientes requisitos: <https://experienceleague.adobe.com/docs/commerce-operations/installation-guide/system-requirements.html>

1.2. Objetivos del Trabajo

El objetivo del proyecto es obtener una arquitectura para *E-Commerce*, extrapolable a cualquier aplicación Web que use una pila de software similar a la presentada, y que permita al administrador modificar la capacidad de esta sin provocar paradas de servicio o minimizándolas al máximo.

Además, es objetivo del trabajo que el coste de esta infraestructura sea mínimo o inexistente en cuanto al software se refiere, y que sea fácilmente trasplantable a otras arquitecturas e incluso nubes públicas. De igual manera se tendrá en cuenta que tanto la arquitectura como el software *E-Commerce*

sean sencillos de administrar desde un punto de vista de negocio como informático.

Dados los requisitos del software de *E-Commerce* Adobe Commerce Magento *Open source*, versión 2.4.5, será necesario crear una arquitectura distribuida que contenga los elementos atómicos fundamentales de forma que cada uno de ellos pueda administrarse independientemente, lo que a su vez permitirá crear grupos de gestión de $n+1$ instancias de cada uno de estos elementos atómicos formando así grupos de escalado que más adelante serían susceptibles de automatizar mediante Dev-Ops.

Comenzando por el intérprete de los scripts de PHP que confirman el núcleo del *E-Commerce* será necesario proveer de intérpretes PHP-FPM versión 8.1 que su vez estarán separados en dos grupos no dependientes para ejecutar tanto la tienda online como el backoffice de la misma, de forma que, aunque la tienda se encuentre en mantenimiento el acceso y administración siga siendo posible.

Para almacenar la transaccionalidad del *E-Commerce*, se creará un clúster Activo-Pasivo de MySQL 8.0 (O MariaDB 10.4) que proporcionará la tolerancia a fallos y la alta disponibilidad deseada. Constará de un servidor Activo primario, sobre el cual se ejecutarán todas las transacciones DDL, DCL y TCL, además de las DML que modifiquen datos (INSERT, UPDATE, DELETE) y $n+1$ servidores Pasivos, secundarios, con mecanismo de replicación que asistirán en modo de sólo lectura al servidor primario con las consultas de tipo DML que no modifiquen datos (SELECT).

Para el guardado y recuperación de objetos, el *E-Commerce* usa una base de datos clave-valor residente en memoria bajo Redis, para este servicio se creará un clúster de $1+n$ servidores con al menos dos bases de datos diferenciadas para la caché de *E-Commerce* y la caché de carritos de usuario que requiere el *E-Commerce* Adobe Commerce Magento *Open source*, la versión compatible de Redis es la 6.2.

Para una interacción más completa y potente con el usuario, será necesario construir un clúster Elasticsearch, el cual proporcionará que las búsquedas de texto generen resultados realmente relevantes a los usuarios del *E-Commerce*. Tal clúster constará también de $1+n$ máquinas independientes las cuales proporcionarán el servicio, ejecutando Elasticsearch versión 7.17.

Finalmente, toda esta infraestructura será presentada al mundo a través de un grupo de servidores Web basados en Nginx, junto con el sistema de caché Varnish, en versiones 1.18 y 7.0 respectivamente.

Para el almacenamiento, tanto de los scripts del *E-Commerce*, como los medios audiovisuales pasando por configuraciones y almacenes persistentes de información, se creará un clúster de almacenamiento basado en GlusterFS, con toda su infraestructura subyacente necesaria.

1.3. Enfoque y método seguido

Se enfocará el proyecto desde diversos puntos de vista diferentes, tratando de obtener el máximo rendimiento, la mayor portabilidad y sobre todo el menor coste posible.

De este modo, desde el punto de vista del rendimiento, el sistema correrá sobre sistemas operativos Linux, virtualizados utilizando contenedores dónde la independencia entre máquina física y contenedor virtualizado lo permita, y además máquinas virtuales dónde el software utilizado lo requiera o justifique de algún modo.

Los ejemplos más claros los tenemos en la gestión del almacenamiento, que será adecuado, aunque no obligatorio, utilizar máquinas virtuales, debido a que se trabajará con un componente crítico, el disco, sobre el cuál es más fácil trabajar en volúmenes virtuales, para máquinas virtuales, que usando montajes anidados en contenedores. Además, la independencia de las máquinas virtuales permite mover los volúmenes de almacenamiento de una manera mucho más efectiva que los puntos de montaje anidados de los contenedores.

Por otro lado, tenemos el ejemplo de los intérpretes, o trabajadores de PHP, que únicamente tendrán que ejecutar un servicio intérprete de PHP para devolver a los servidores web, por lo que son candidatos excelentes para utilizar contenedores.

Desde el punto de vista de la portabilidad, las máquinas y contenedores se configuran lo más atómicamente posible de forma que su configuración pueda ser replicada con facilidad en cualquier otra arquitectura, ya sea basada en diferente hipervisor, *on-premises*, en una nube pública, etc. O incluso devolverla a un sistema monolítico directamente sobre hardware.

Desde un punto de vista de costes, todo el software utilizado es libre y gratuito, si bien existe la posibilidad de adquirir suscripciones de mantenimiento en muchas de las dependencias de software, como el hipervisor Proxmox, los SGBD MariaDB y Redis, pasando por el servidor Web Nginx.

1.4. Planificación del Trabajo

El proyecto es tecnológicamente viable, en tanto en cuanto se basa en sistemas operativos, hipervisores y un software *E-Commerce* libres y en ciclo de vida.

Todos los componentes atómicos a su vez cuentan con una gran base de conocimientos y grandes comunidades sobre las que apoyarse tanto en el despliegue de la solución, como en configuración y documentación.

El hardware sobre el que se desplegará la infraestructura virtual está disponible y no se necesita de ningún tipo de licencia para su uso, aunque debido a que se trata de hardware de propósito general y que únicamente se dispone de un equipo sobre el que ejecutar la infraestructura, quedan fuera de

alcance pruebas o métricas de rendimiento de esta, en tanto en cuanto estas estarían claramente limitadas por el hardware.

La planificación general se ha compuesto de los 4 hitos principales de la asignatura de TFG en el campus UOC, los cuales se han esquematizado usando un diagrama de Gantt mediante la metodología de proyecto en cascada, por lo que cada uno de los hitos y se debe realizar respetando el orden de planificación y con una capacidad limitada para superponer tareas, algo especialmente adaptado a una planificación de trabajo para una sola persona.

Cada hito corresponde con una entrega mediante el registro de evaluación continua del campus y a su vez está dividido en tareas concretas, los hitos de planificación son los siguientes:

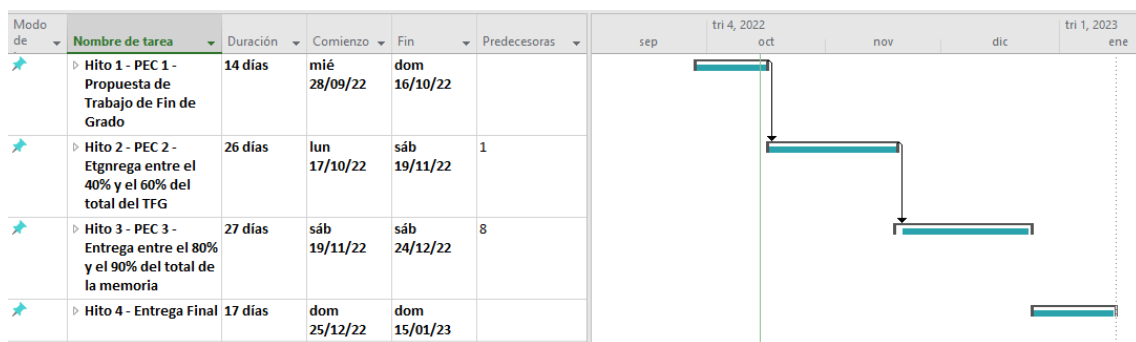


Ilustración 1 – Planificación de TFG.

Hito 1. Propuesta de plan de trabajo.

En este primer hito de planificación se crea la propuesta del plan de trabajo, que ha de incluir entre otras cosas la descripción del proyecto, su justificación y planificación de objetivos, además de su título.

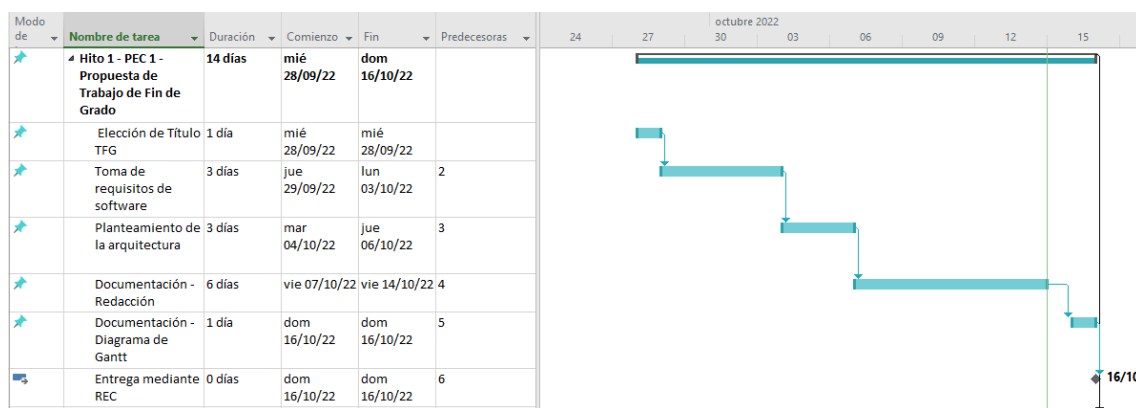


Ilustración 2 - Hito 1. Propuesta de plan de trabajo.

Hito 2. Entrega entre el 40% y el 60% del total del TFG.

En este hito se provisiona el hardware necesario que ejecutará el hipervisor que a su vez permitirá virtualizar las máquinas y contenedores, los

cuales alojarán los servicios distribuidos sobre los que correrá el software *E-Commerce Adobe Commerce Magento Open source*.

También se prepararán las plantillas que se usarán, descargándose del repositorio público de plantillas de Proxmox en el caso de los contenedores y creando una máquina virtual de plantilla en el caso de las máquinas virtuales.

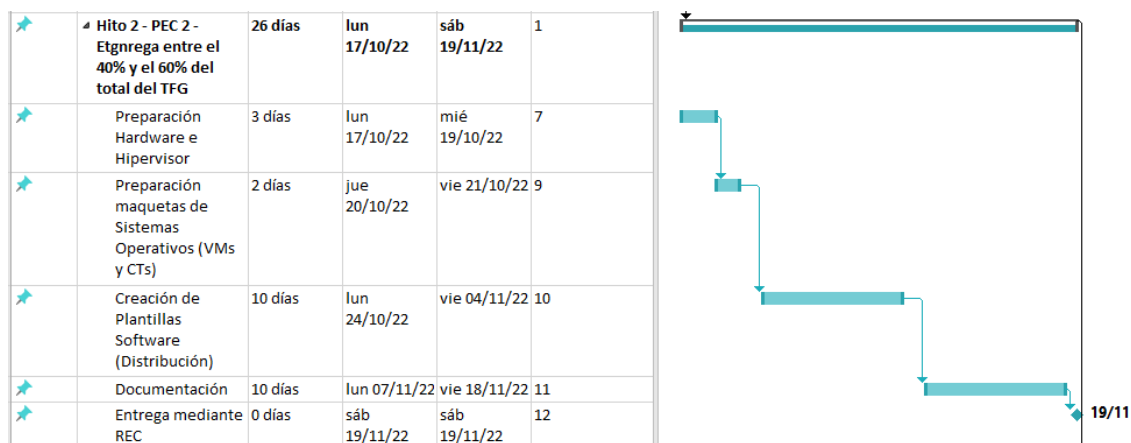


Ilustración 3 - Hito 2. Entrega entre el 40% y el 60% del total del TFG

Hito 3. Entrega entre el 80% y 90% del total de la memoria.

En este intervalo se realizará la planificación de la red local privada mediante la que se comunicarán todas las partes distribuidas de la arquitectura, se implementará dicha arquitectura e instalará el software *E-Commerce Adobe Commerce Magento Open source*, tras una serie de pruebas de funcionamiento se implementará la alta disponibilidad de la arquitectura y su posterior documentación.

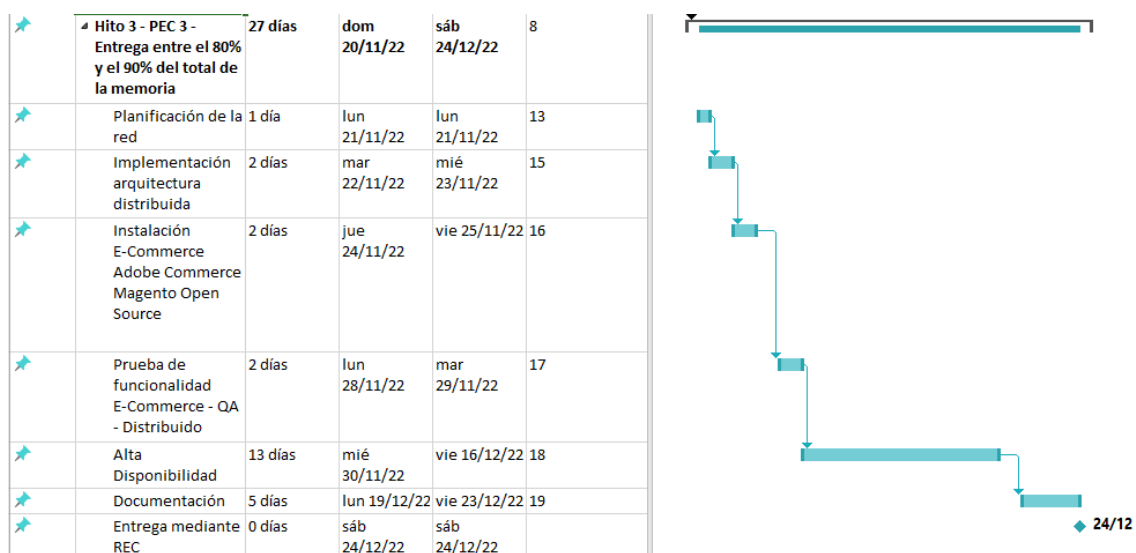


Ilustración 4 - Hito 3. Entrega entre el 80% y 90% del total de la memoria.

Hito 4. Entrega final.

En este último hito se realizará la prueba conceptual del software *E-Commerce* en alta disponibilidad, además de la revisión de la memoria,

preparación de la presentación, su defensa y entrega final de la memoria del trabajo de fin de grado.

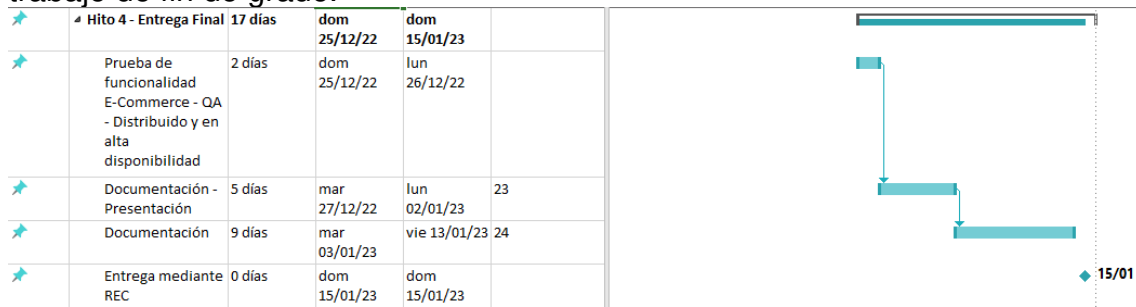


Ilustración 5 - Hito 4. Entrega final.

1.5. Breve resumen de productos obtenidos

Al finalizar el trabajo, se obtendrá una arquitectura de virtualización basada en software libre, gratuita, altamente escalable y con la capacidad de ejecutar de forma segura y eficiente cualquier aplicación Web basada en un stack LEMP, concretamente Adobe Commerce Magento *Open source*.

1.6. Breve descripción de los otros capítulos de la memoria

A continuación, una breve descripción del contenido de los capítulos siguientes:

En el capítulo 2 se hará un breve repaso del marco del que nace la necesidad que se trata de satisfacer en este trabajo de fin de grado. Se definirán los conceptos de *E-Commerce*, los sistemas monolíticos, la distribución de sistemas, las variantes de dicha distribución y finalmente cómo la virtualización redonda en todo lo anterior para ofrecer alta disponibilidad y tolerancia a fallos.

En el capítulo 3 se detallará la arquitectura necesaria para realizar el trabajo, partiendo del hardware necesario para ello, la configuración de red y almacenamiento, copias de seguridad y qué máquinas de infraestructura son necesarias para simular un entorno productivo.

El capítulo 4 desarrolla el despliegue de un *stack* LEMP, además del resto de servicios necesarios para poder instalar Adobe Commerce Magento *Open source*, todo ello partiendo de una base distribuida por servicios.

En el capítulo 5 se detallan las configuraciones necesarias para llevar el servicio de E-Commerce a un nivel tolerante a fallos y redundante.

El capítulo 6 se cierra el trabajo con una revisión de las conclusiones obtenidas durante el desarrollo del trabajo.

El capítulo 7 contiene un glosario de los términos utilizados durante la documentación del trabajo.

El capítulo 8 contiene la bibliografía utilizada durante la documentación del trabajo.

El capítulo 9 son una serie de anexos, entre los cuales se encuentran los archivos de configuración de todo el software necesario.

2. Marco del trabajo

En este capítulo repasaremos los elementos clave, desde un punto de vista definitorio, que conforman el contexto sobre el que se realiza este trabajo de fin de grado.

2.1. Comercio electrónico

El comercio electrónico o *E-Commerce*, término acuñado ya por el año 1984 en California por los miembros del *California State Assembly's Utilities & Commerce Committee*, se define como cualquier forma de transacción o intercambio de información comercial que use como canal de transmisión de datos una o varias redes de telecomunicación, hoy en día prácticamente todo vía web.

E-Commerce es término muy amplio que engloba muchos tipos de comercio diferentes, de forma habitual se definen cinco tipos diferentes de comercio electrónico en función de los actores que forman parte de la transacción comercial los cuales quedan definidos como:

- B2B, *Bussiness to Bussiness*: el comercio electrónico entre empresas, ya sean mayoristas, o que formen parte de la cadena de transformación de materias. Este tipo de comercio electrónico se aprovecha especialmente de la globalización del comercio electrónico.
- B2C, *Bussines to Consumer*: el caso específico de este trabajo de fin de grado se trata de un tipo de comercio que usa normalmente la Web para las transacciones, ya sean compraventa de productos, servicios o subastas online.
- A2A, *Administration 2 Administration*: es un tipo de comercio electrónico que tiene lugar entre dos administraciones del estado, en el caso del estado español, tendrían lugar como A2A todas las transacciones entre administraciones central y autonómicas, por ejemplo.
- B2A, *Bussiness 2 Administration*: este tipo de comercio electrónico engloba las transacciones comerciales entre una empresa y una administración. Especialmente importante para acelerar los trámites administrativos relacionados.
- C2C, *Consumer 2 Consumer*: esta variación de comercio electrónico se basa en la comunicación entre dos usuarios para efectuar compraventa

de artículos, en este caso particular tenemos tanto las webs de subastas como las plataformas de compraventa de segunda mano.

Este trabajo trata particularmente de un sistema de comercio electrónico del tipo B2C o *Business to Consumer*, concretamente de una tienda online, es decir, un portal web dónde la empresa pone en venta sus productos a usuarios particulares para que estos los adquieran vía internet.

Como se verá en próximos capítulos de esta memoria, el uso de equipamiento informático tradicional para alojar tiendas Online conlleva una serie de problemas asociados, específicamente en materia de rendimiento y capacidad de respuesta, los cuales se intentarán paliar en la medida de lo posible usando técnicas más modernas en informática, tales como la virtualización y la distribución de aplicaciones web.

2.2. Sistemas monolíticos

Los sistemas monolíticos, quizás aplicaciones monolíticas sea más acertado en este contexto, se caracterizan por tener todas sus funcionalidades agrupadas en un único ente central, en este se encuentran instalados todos los softwares necesarios para el funcionamiento de la aplicación, así como todos los archivos y configuraciones. Son el modelo usado para desarrollar las primeras aplicaciones y sistemas operativos, cuando los ordenadores, especialmente los servidores, eran prohibitivamente caros y se intentaba extraer el máximo rendimiento posible.

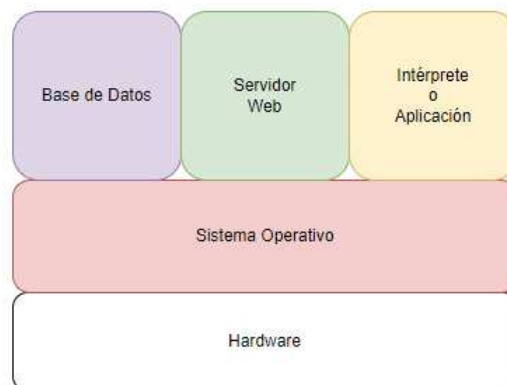


Ilustración 6 - Ejemplo gráfico de sistema monolítico.

Esta característica le confiere ciertas ventajas positivas, tales como facilidad de desarrollo, despliegue y facilidad de ejecución, al estar confinadas en el mismo contexto, además, debido a la simplicidad de su estructura son menos costosas de desarrollar y desde un punto de vista de sistemas, más sencillas de mantener y asegurar, tanto desde el punto de vista de la ciberseguridad como desde la resiliencia.

Sin embargo sus puntos débiles también son importantes en contraposición a su simpleza, una aplicación monolítica tiene muy poca o nula adaptabilidad a las variaciones de uso, o la cantidad de trabajo que puede

procesar, al igual que en los sistemas por lotes de las primeras etapas de la informática moderna, el ritmo de trabajo lo marca la capacidad de la aplicación, o del *hardware* subyacente que la ejecuta, esto en el estado actual donde prima la capacidad de servicio, es un aspecto absolutamente a evitar. Además, la dependencia del hardware provoca que la avería de un único componente puede tener el sistema parado horas o días, lo cual significará con prácticamente total seguridad, pérdidas económicas.

En el contexto de este trabajo, una tienda online de venta a consumidor B2C, un sistema monolítico tiene serias carencias a la hora de servir cuando la variabilidad de usuarios es muy alta. Más aún, aumentar el rendimiento de un sistema monolítico implica una parada de servicio para ampliar cual fuese el elemento que provocase el tope de rendimiento, como CPU, RAM, almacenamiento, e incluso en algunos casos ancho de banda de red disponible. Estas paradas de servicio conllevan, además de la propia pérdida de la capacidad de interactuar con los consumidores, una serie de peligros inherentes al apagado total del hardware, peligros que son directamente proporcionales a la antigüedad del hardware en cuestión.

2.3. Sistemas distribuidos

De los múltiples problemas de adaptabilidad de los sistemas monolíticos del punto anterior surge la necesidad de distribuir las aplicaciones, esto es, generando distintas instancias de estas, lo que nos permite balancear la carga de la aplicación entre instancias, dando a su vez alta disponibilidad y permitiendo escalar horizontalmente las instancias por separado evitando paradas de servicio.

Una forma completamente válida de distribución de sistemas, consiste en multiplicar los sistemas monolíticos en $n+1$ sistemas idénticos sobre los que se distribuirá la carga de la aplicación, si bien a priori parece una opción simple y fácilmente realizable, supone una serie de retos bastante complicados de alcanzar, tales como el guardado de la información de forma centralizada, que puede requerir de una gran inversión en una infraestructura de almacenamiento en red, o los propios mecanismos de distribución de carga y sincronización entre servidores, el cual puede suponer a su vez agregar otro sistema monolítico delante del actual, echando por tierra gran parte de la ventaja de la distribución.

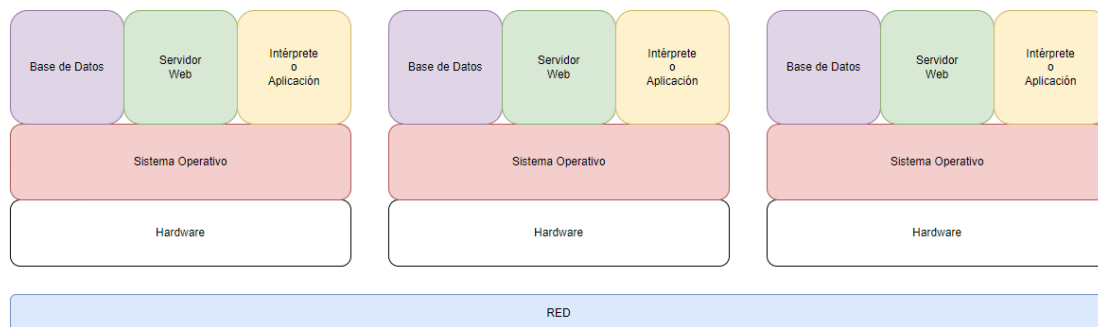


Ilustración 7 - Ejemplo gráfico de sistema distribuido.

La forma principal de evitar los contratiempos de multiplicar sistemas monolíticos, obtener el mayor rendimiento del hardware disponible y de evitar la necesidad de tener que invertir en nuevo, consiste en dividir los sistemas monolíticos en partes menores codependientes para la aplicación, pero que pueden trabajar independientemente siempre y cuando exista una red de comunicación y sincronización entre estas partes.

2.4. Clústeres y Microservicios

La separación granular vista anteriormente nos deja una nube de pequeños sistemas distribuidos que de forma global nos permitirán ejecutar la aplicación deseada, pero que podremos manipular individualmente, pudiendo escalar cada uno de ellos en función de la necesidad de carga del global, haciendo el sistema mucho más eficiente, rápido y tolerante a variaciones en la carga de trabajo.

Toma entonces especial relevancia la comunicación entre servicios para que el todo funcione, algo que es trivial en un sistema basado en nube pública, pero que es especialmente importante en sistemas basados en virtualización *on-premises* y que a su vez ha de ser distribuida y redundada.

La parte comunicativa en el contexto de este trabajo recae en una red de área local estándar mediante IPv4, virtualizada, mientras que los mecanismos de sincronización entre las partes de la aplicación se satisfarán, en parte mediante software que ya está preparado para funcionar en un entorno distribuido, o incorporando softwares intermedios que se encargarán de tal propósito.

Como ejemplo, este Trabajo de Fin de Grado consiste en la instalación del software *E-Commerce* Adobe Magento *Open source*, el cual está programado en PHP y que funciona exclusivamente con sistemas operativos Linux y “pilas” de aplicación web LAMP o LEMP.

Para su correcto funcionamiento depende de cuatro componentes principales:

- Sistema Operativo Linux.
- Servidor Web: Apache o Nginx
- Servidor de bases de datos: MySQL o MariaDB
- Intérprete de PHP

Inicialmente la distribución de la aplicación se llevará a cabo montando servidores específicos para cada uno de estos componentes, estos servidores específicos se podrán pues redundar para obtener alta disponibilidad y tolerancia a fallos.

La redundancia de estos servicios atómicos se obtiene de dos maneras, dependiendo del tipo de servicio, concretamente sobre la necesidad o no de sincronización entre las distintas instancias del servicio. Si el servicio en cuestión debe satisfacer las características A.C.I.D. (**A**tomicity, **C**onsistency,

Isolation and Durability), todas sus instancias deberán estar sincronizadas de alguna forma, esto es para los servicios de BBDD y de cacheado de sesiones, se montará un clúster de instancias intercomunicadas. Por otro lado, tenemos los servicios que no requieren satisfacer estas características, como pueden ser los intérpretes de PHP o los balanceadores, incluso el servicio de *full page cache*, los cuales se considerarán como microservicios y se balancearán mediante los algoritmos más adecuados para cada tipo de tarea.

2.5. Virtualización y contenedores

El último elemento necesario para detallar el contexto sobre el que se desarrolla este trabajo es la virtualización, la cual consiste en utilizar un tipo de sistema operativo especial, llamado hipervisor, instalado sobre el hardware físico, es decir los servidores. Este hipervisor permite a su vez crear distintos sistemas virtuales, cada uno con su propio sistema operativo.

El hipervisor elegido para este trabajo es Proxmox, software libre, comunitario y basado en Debian, KVM y LXC, esto permite por un lado crear máquinas virtuales comunes bajo KVM, pero también contenedores bajo LXC.

Un contenedor es un tipo especial de máquina virtual, la cual tiene un sistema de archivos en un directorio del servidor, sus características de memoria, procesador y ciertas opciones, pero que comparte el núcleo central con el hipervisor, lo cual le confiere de gran flexibilidad en la configuración, pero le resta en dependencia. Aun así, se integrarán tanto máquinas virtuales como contenedores en este trabajo, en función de cómo encajen y beneficien las características de cada tipo de instancia con los servicios que se han de configurar.

Una máquina virtual es una emulación completa de una computadora, la cual tendrá su propio espacio de almacenamiento, comúnmente un disco virtual en un archivo o un disco en red en una cabina de almacenamiento tiene su propio kernel funcionando en su propio espacio de trabajo a nivel de usuario. Las máquinas virtuales tienen un aislamiento total del sistema anfitrión lo que permite instalar otros sistemas operativos no Linux, así como añadir discos virtuales extra, y asignarles directamente dispositivos a nivel de hardware, tales como tarjetas gráficas, aceleradores de IA, tarjetas de red, etc.

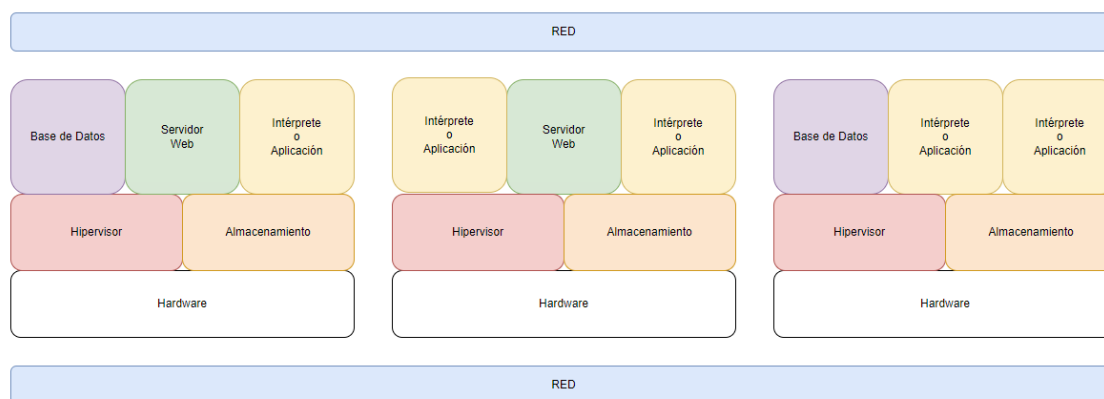


Ilustración 8 - Ejemplo gráfico de infraestructura virtual.

3. Arquitectura

3.1. Hardware e Hipervisor

La arquitectura del sistema, como ya se adelantó en anteriores capítulos funciona sobre un hipervisor de virtualización Proxmox 7 sobre Debian 11, ejecutándose sobre hardware de propósito general, sobre un único servidor:

- Placa Base: Asus Z170i
- Procesador: Intel Core i7 6700k
- Memoria RAM: Gskill 32Gb DDR4@3200Mts
- Disco Duro principal: Toshiba KBG30ZMS256G NVMe 256Gb
- Disco Duro secundario: Crucial MX500 Sata3

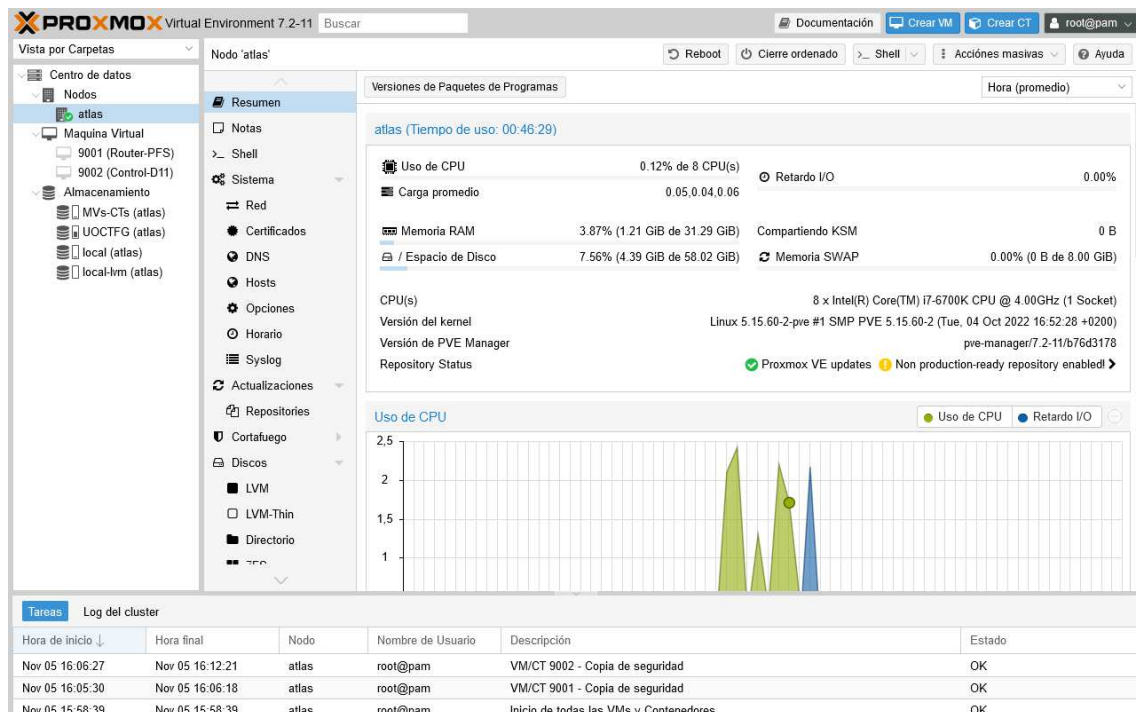


Ilustración 9 – Proxmox VE (Virtual Environment).

3.2. Configuración de Red

En este trabajo se emula un sistema virtual aislado de internet, con un único acceso externo mediante un router con NAT y redirección de puertos, como se haría en una instalación on-premises o en una nube pública por una PYME.

El servidor tiene únicamente una tarjeta de red cableada, por lo que la configuración de red interna se realiza a través de un switch virtual de tipo puente que permite la comunicación interna entre todas las máquinas. No habiendo involucrada ninguna interfaz física, este enlace funcionará en todas las máquinas y contenedores a 10Gbps, minimizando las latencias y maximizando la velocidad de comunicación entre servicios.

La interfaz cableada se conecta a una red LAN doméstica, también con una conexión de puente la cual tiene asignada la interfaz de red física correspondiente, esto permite exponer el interfaz web del hipervisor a la red local, además de permitir asignar interfaces de máquinas virtuales y exponerlas a la red LAN para simular un acceso a internet.

Todas las máquinas y contenedores pertenecientes a la infraestructura tienen una IP interna del rango de clase C 10.20.22.0/24, otorgadas por la máquina *router* de forma estática según la dirección MAC de la máquina o contenedor en cuestión. El *router*, a su vez tiene una IP interna del rango 192.168.5.0/24 que actúa como IP externa y puerta de enlace para el resto de las máquinas y contenedores instalados.

Junto con el servicio de DHCP también se presta el servicio de DNS para la red interna, de forma que cada máquina es resuelta a su IP correspondiente internamente y además permite la creación de punteros tipo A hacia grupos de máquinas concretas para servicios concretos, con fin de proporcionar cierto grado de tolerancia a fallos, i.e. el servicio de almacenamiento *str* apunta a las tres máquinas de almacenamiento STR1-D11, STR2-D11 y STR3-D11.

Nombre ↑	Tipo	Activo	VLAN ...	Puertos/Es...	CIDR	Puerta de enlace	Comentario
enp0s31f6	Dispositivo de ...	Si	No				
vmbr0	Linux Bridge	Si	No	enp0s31f6	192.168.5.40/24	192.168.5.240	WAN
vmbr1	Linux Bridge	Si	No		10.20.22.0/24		LAN
wlp5s0	Desconocido	No	No				

Log del cluster					
Hora de inicio ↓	Hora final	Nodo	Nombre de Usuario	Descripción	Estado
Dic 07 04:57:43	Dic 07 04:57:45	atlas	root@pam	Actualización de paquetes en la Base de Datos	OK
Dic 07 01:00:01	Dic 07 01:05:29	atlas	root@pam	Tarea de copia de seguridad	OK

Ilustración 10 - Proxmox VE. Configuración de Red.

3.3. Configuración de Almacenamiento

El almacenamiento a nivel físico consiste en un único disco duro SSD SATA de 500Gb de capacidad, en el cual se guardan todos los discos virtuales, tanto de máquinas y contenedores, dado que no se trata de un entorno productivo no corresponde la necesidad de redundancia en el almacenamiento, tanto por costes, como por complejidad añadida en la infraestructura física.

Por el contrario, el almacenamiento de la infraestructura virtual sí está redundado, por triplicado, y se ha configurado mediante el uso del sistema de

archivos distribuido GlusterFS tal y como se detalla en el apartado 4.2 de esta memoria.

3.4. Configuración de Copias de Seguridad

Como medida de precaución a la posibilidad de tener que reconfigurar un servidor, corrupción de un servicio, o para poder escalar horizontalmente algún servicio, se configura un servicio de copia de seguridad, incluido en el propio hipervisor, contra un servidor Proxmox Backup Server, ya existente en la red local, al cual se le añade un repositorio específico, o *datastore*, en el que se guardarán de forma automática una vez al día, o de forma manual a voluntad copias de cualquiera de las máquinas virtuales o contenedores de la infraestructura.

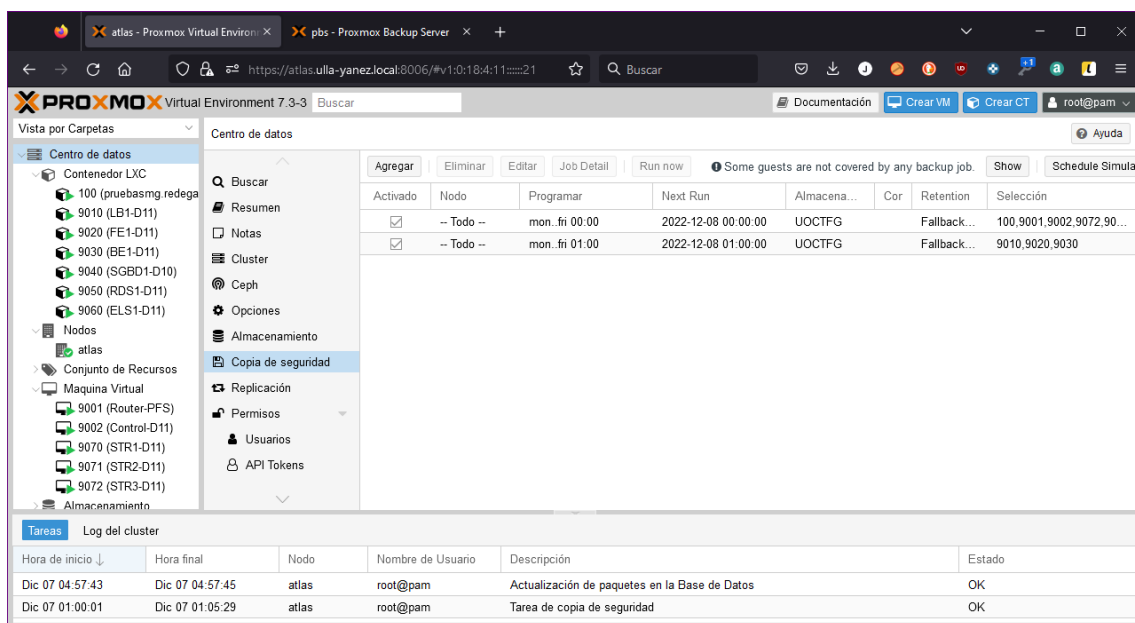


Ilustración 11 - Proxmox VE. Programación copias de seguridad.

El uso de este servicio confiere la capacidad de salvaguardar una máquina o contenedor en cuestión de minutos, así como la de restaurar dicha máquina y estado sobre escribiendo la original o bien sobre una instancia nueva permitiendo generar clones preconfigurados, prácticamente listos para trabajar.

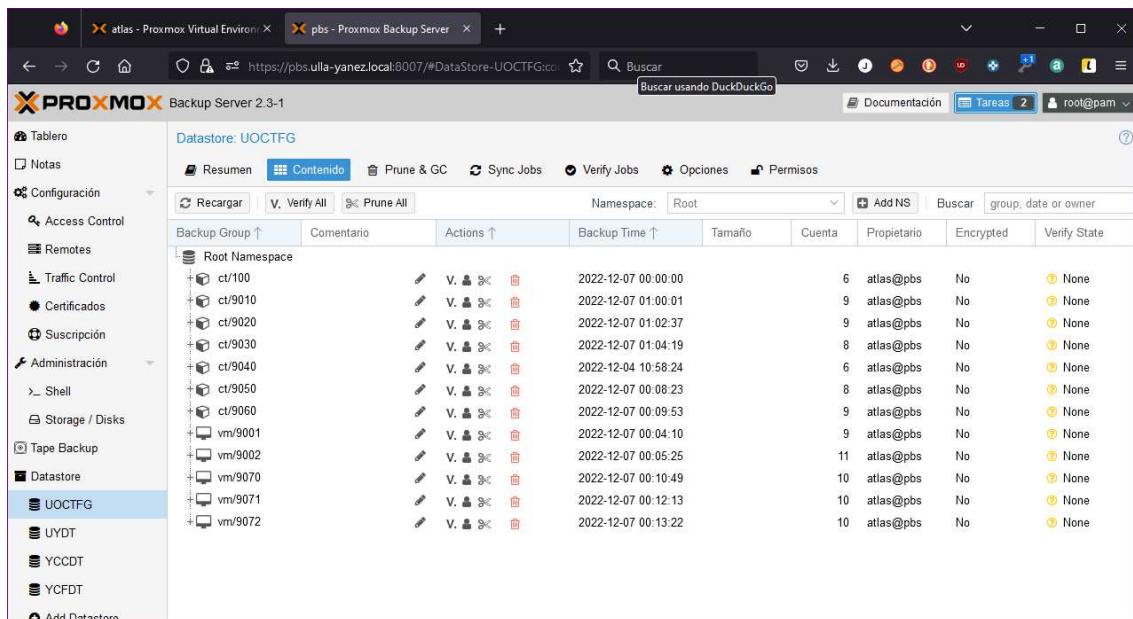


Ilustración 12 - Proxmox Backup Server. Datastore UOCTFG.

3.5. Máquinas de Infraestructura

Para que la infraestructura funcione correctamente y con fin de emular un sistema totalmente aislado de internet, se precisan dos máquinas virtuales extra:

Una máquina router/firewall basado en PFSense cuya labor será conectar la red externa con la interna, redirigiendo los puertos HTTP 80, HTTPS 443 al proxy reverso, así como el puerto SSH 22 hacia la máquina control

La segunda máquina extra necesaria es Control, una máquina destinada al gobierno de la infraestructura, necesaria para programar las posibles tareas de mantenimiento de la aplicación, además de servir como puente exterior para la conexión de los administradores, tanto de la infraestructura como del *E-Commerce*.

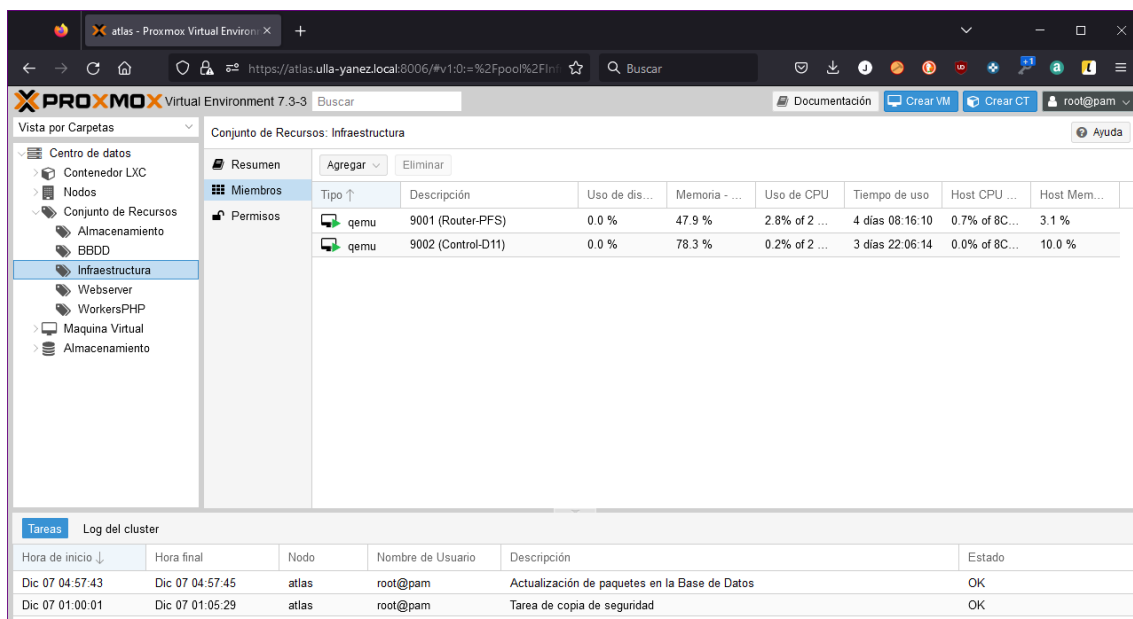


Ilustración 13 - Proxmox VE. Pool de máquinas de infraestructura.

4. Software

Sobre el hardware y la arquitectura virtual descrita en el apartado anterior se ejecutarán las máquinas virtuales y contenedores que nos permitirán ejecutar a su vez el software necesario para poder ejecutar finalmente el software de e-commerce, que al final no es más que la conjunción de todos los elementos anteriores y el objetivo del trabajo.

Para ello, en este capítulo se detallan la instalación y configuración de cada uno de los componentes software que formarán parte de la infraestructura distribuida y redundada que ejecutara el software *E-Commerce*.

El desarrollo de este capítulo se hará en un orden específico para la construcción de la aplicación, preparando en primer lugar los almacenes de datos y servidores de Bases de Datos, para continuar con los accesorios, los intérpretes de PHP, la caché y servidor web para finalizar con la propia instalación del software de e, Adobe *E-Commerce* *Magento Open source*.

4.1. Sistemas Operativos

Continuando con el objetivo de crear una plataforma totalmente basada en software libre, se escoge como sistema operativo para todas las máquinas y contenedores que conforman la arquitectura el sistema operativo Linux Debian.

En prácticamente todos los casos se utilizará la última versión estable de la distribución disponible en este momento: Debian 11 *Bullseye*, sin embargo, por motivos de compatibilidad con el software SGBD, en el caso específico de los contenedores que ejecutarán el motor de bases de datos, se utilizará la versión estable anterior Debian 10 *Buster*, la cual aún cuenta con mantenimiento de seguridad por la comunidad.

4.2. GlusterFS

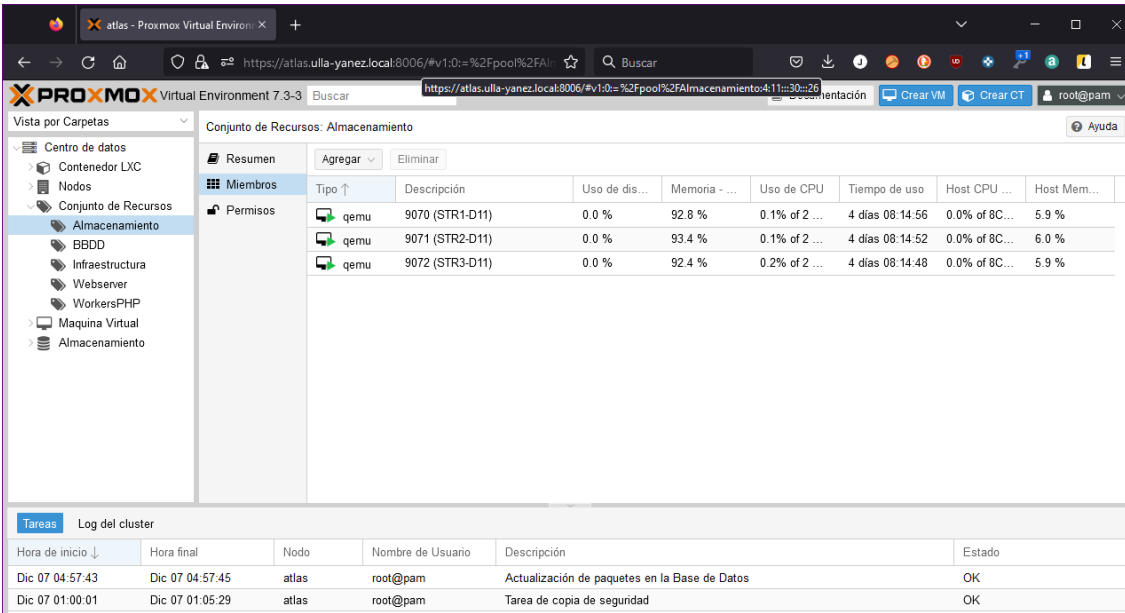
Para proveer de almacenamiento a la aplicación se provee un clúster de almacenamiento distribuido y redundado basado en GlusterFS.

GlusterFS es un sistema de almacenamiento en red escalable, creado por la empresa Gluster Incorporated, adquirida posteriormente por Red Hat. GlusterFS es un sistema de archivos apropiado para aplicaciones con interacción intensiva de datos, como puede ser un software de *E-Commerce* ya que se diseñó para utilizar el espacio de usuario, por lo que no sufre de pérdidas de rendimiento en llamadas al sistema.

Para esta infraestructura se crea un clúster de almacenamiento sobre tres servidores en máquinas virtuales sobre los cuales se replicará el sistema de archivos dónde se guardará el *E-Commerce*, esto permitirá a priori aprovechar al máximo el uso de disco local en entornos on-premises o posibilitar la replicación de los datos en varias zonas de disponibilidad en virtualización en nube pública.

La instalación de GlusterFS es muy simple en Debian 11, ya que es un paquete disponible en los repositorios oficiales de la distribución, sólo hay que tener en cuenta que para que el clúster pueda sobrevivir a un evento disruptivo en DNS, deberá configurarse en el archivo local de hosts la resolución correspondiente para todos los miembros del clúster.

El sistema de archivos distribuido en red se instala en tres máquinas virtuales en la infraestructura, STR1-D11, STR2-D11 y STR3-D11. Estas máquinas virtuales se crean con dos discos virtuales, uno para el sistema y otro, de 30Gb, dónde se configurará GlusterFS para la replicación de la información.



Tipo	Descripción	Uso de dis...	Memoria - ...	Uso de CPU	Tiempo de uso	Host CPU ...	Host Mem...
qemu	9070 (STR1-D11)	0.0 %	92.8 %	0.1% of 2 ...	4 días 08:14:56	0.0% of 8C...	5.9 %
qemu	9071 (STR2-D11)	0.0 %	93.4 %	0.1% of 2 ...	4 días 08:14:52	0.0% of 8C...	6.0 %
qemu	9072 (STR3-D11)	0.0 %	92.4 %	0.2% of 2 ...	4 días 08:14:48	0.0% of 8C...	5.9 %

Hora de inicio	Hora final	Nodo	Nombre de Usuario	Descripción	Estado
Dic 07 04:57:43	Dic 07 04:57:45	atlas	root@pam	Actualización de paquetes en la Base de Datos	OK
Dic 07 01:00:01	Dic 07 01:05:29	atlas	root@pam	Tarea de copia de seguridad	OK

Ilustración 14 - Proxmox VE. Pool de máquinas de almacenamiento.

Se añade la siguiente información al archivo hosts de cada uno de los servidores:

```
10.20.22.70 str1 STR1-D11
10.20.22.71 str2 STR2-D11
10.20.22.72 str3 STR3-D11
```

A continuación, se instala el software de GlusterFS en todos los servidores:

```
root@STR1-D11:~# apt install glusterfs-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  attr fuse glusterfs-client glusterfs-common keyutils libaio1 libevent-2.1-7 libgfapi0 libgfchangelog0 libgfrpc0 libgfxdr0 libglusterd0
  libglusterfs0 libinih1 libnfsidmap2 liburcu6 nfs-common psmisc python3-cffi-backend python3-cryptography python3-jwt
  python3-prettytable rpcbind xfsprogs
Paquetes sugeridos:
  open-iscsi watchdog python-cryptography-doc python3-cryptography-vectors python3-crypto xfsdump acl quota
Se instalarán los siguientes paquetes NUEVOS:
  attr fuse glusterfs-client glusterfs-common glusterfs-server keyutils libaio1 libevent-2.1-7 libgfapi0 libgfchangelog0 libgfrpc0
```

Ilustración 15 - Instalación de GlusterFS 1.

Se crea la partición y el sistema de archivos:

```
root@STR1-D11:~# fdisk /dev/sdb

Bienvenido a fdisk (util-linux 2.36.1).
Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.
Tenga cuidado antes de utilizar la orden de escritura.

El dispositivo no contiene una tabla de particiones reconocida.
Se ha creado una nueva etiqueta de disco DOS con el identificador de disco 0xdbc3d803.

Orden (m para obtener ayuda): n
Tipo de partición
  p  primaria (0 primaria(s), 0 extendida(s), 4 libre(s))
  e  extendida (contenedor para particiones lógicas)
Seleccionar (valor predeterminado p):

Se está utilizando la respuesta predeterminada p.
Número de partición (1-4, valor predeterminado 1):
Primer sector (2048-67108863, valor predeterminado 2048):
Último sector, +/-sectores o +/-tamaño{K,M,G,T,P} (2048-67108863, valor predeterminado 67108863):

Crea una nueva partición 1 de tipo 'Linux' y de tamaño 32 GiB.

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.

root@STR1-D11:~#
```

Ilustración 16 - Instalación de GlusterFS 2.

```

root@STR2-D11:~# mkfs.ext4 -L GFS1 /dev/sdb1
mke2fs 1.46.2 (28-Feb-2021)
Discarding device blocks: done
Creating filesystem with 8388352 4k blocks and 2097152 inodes
Filesystem UUID: 9dd538c9-85be-4149-b1c6-9ad71c722259
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 16056
    32, 2654208,
    4096000, 7962624

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

```

Ilustración 17 - Instalación de GlusterFS 3.

A continuación se crean los directorios para los *bricks*, pedazos de almacenamiento distribuido dónde se guardarán los datos. Estos bricks se crean en espacio de usuario, convenientemente se añade la información de automontaje al inicio para el nuevo almacenamiento:

```

root@STR2-D11:~# mkdir -p /gluster/bricks/2
root@STR2-D11:~# blkid
/dev/sdb1: LABEL="GFS1" UUID="9dd538c9-85be-4149-b1c6-9ad71c722259"
    BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="d7bbe329-01"
/dev/sda1: UUID="e78db844-617a-4959-82cd-cabfdada4341" BLOCK_SIZE="
4096" TYPE="ext4" PARTUUID="d810fe94-01"
/dev/sda2: UUID="3fabec51-99dc-48f9-8f33-75ea8bb7cdd3" TYPE="swap"
    PARTUUID="d810fe94-02"
root@STR2-D11:~# echo 'UUID=9dd538c9-85be-4149-b1c6-9ad71c722259 /g
luster/bricks/2 ext4 defaults 0 1' >> /etc/fstab
root@STR2-D11:~# mount -a
root@STR2-D11:~#

```

Ilustración 18 - Instalación de GlusterFS 4.

Se utiliza el UUID de la partición para evitar problemas en el arranque, de la máquina virtual, esto nos permitiría migrar la máquina entre diferentes servidores, e incluso infraestructuras sin perder funcionalidad.

Para los tres servidores tenemos entonces configurado el almacenamiento:

```

root@STR1-D11:~# df -h

```

S.ficheros	Tamaño	Usados	Disp	Uso%	Montado en
udev	976M	0	976M	0%	/dev
tmpfs	199M	468K	198M	1%	/run
/dev/sda1	13G	1,6G	11G	13%	/
tmpfs	992M	0	992M	0%	/dev/shm
tmpfs	5,0M	0	5,0M	0%	/run/lock
tmpfs	199M	0	199M	0%	/run/user/0
/dev/sdb1	32G	24K	30G	1%	/gluster/bricks/1

Ilustración 19 - Instalación de GlusterFS 5.

```

root@STR2-D11:~# df -h
S.ficheros      Tamaño Usados  Disp Uso% Montado en
udev            976M      0  976M  0% /dev
tmpfs           199M    464K  198M  1% /run
/dev/sda1       13G     1,6G   11G  13% /
tmpfs           992M      0  992M  0% /dev/shm
tmpfs           5,0M      0   5,0M  0% /run/lock
tmpfs           199M      0  199M  0% /run/user/0
/dev/sdb1       32G     24K   30G  1% /gluster/bricks/2

```

Ilustración 20 - Instalación de GlusterFS 6.

```

root@STR3-D11:~# df -h
S.ficheros      Tamaño Usados  Disp Uso% Montado en
udev            976M      0  976M  0% /dev
tmpfs           199M    464K  198M  1% /run
/dev/sda1       13G     1,6G   11G  13% /
tmpfs           992M      0  992M  0% /dev/shm
tmpfs           5,0M      0   5,0M  0% /run/lock
tmpfs           199M      0  199M  0% /run/user/0
/dev/sdb1       32G     24K   30G  1% /gluster/bricks/3

```

Ilustración 21 - Instalación de GlusterFS 7.

El siguiente paso será comprobar en el servidor STR1-D11 que se puede comunicar con el resto de miembros del cluster STR2-D11 y STR3-D11:

```

root@STR1-D11:~# gluster peer probe str2
peer probe: success
root@STR1-D11:~# gluster peer probe str3
peer probe: success
root@STR1-D11:~# |

```

Ilustración 22 - Instalación de GlusterFS 8.

Para terminar creando el espacio de almacenamiento distribuido y triplemente redundado en red:

```

root@STR1-D11:~# gluster volume create ecommerce replica 3 str1:/gluster/bricks/1
/brick str2:/gluster/bricks/2/brick str3:/gluster/bricks/3/brick
volume create: ecommerce: success: please start the volume to access data
root@STR1-D11:~# |

```

Ilustración 23 - Instalación de GlusterFS 9.

```

root@STR1-D11:~# gluster volume list
ecommerce
root@STR1-D11:~# gluster volume start ecommerce
volume start: ecommerce: success
root@STR1-D11:~# |

```

Ilustración 24 - Instalación de GlusterFS 10.

Se comprueba el correcto funcionamiento del volumen, y se añade la configuración de red que permita al resto de máquinas del segmento de red, acceder al nuevo almacenamiento:

```

root@STR1-D11:~# gluster volume status ecommerce
Status of volume: ecommerce
Gluster process                TCP Port  RDMA Port  Online  Pid
-----
Brick str1:/gluster/bricks/1/brick  49152    0          Y      5601
Brick str2:/gluster/bricks/2/brick  49152    0          Y      5427
Brick str3:/gluster/bricks/3/brick  49152    0          Y      5448
Self-heal Daemon on localhost      N/A      N/A        Y      5618
Self-heal Daemon on str3           N/A      N/A        Y      5465
Self-heal Daemon on str2           N/A      N/A        Y      5444

Task Status of Volume ecommerce
-----
There are no active volume tasks

```

Ilustración 25 - Instalación de GlusterFS 11.

```

root@STR1-D11:~# gluster volume info ecommerce
Volume Name: ecommerce
Type: Replicate
Volume ID: 1478013c-35f8-4528-b78a-3090fa70b2d4
Status: Started
Snapshot Count: 0
Number of Bricks: 1 x 3 = 3
Transport-type: tcp
Bricks:
Brick1: str1:/gluster/bricks/1/brick
Brick2: str2:/gluster/bricks/2/brick
Brick3: str3:/gluster/bricks/3/brick
Options Reconfigured:
cluster.granular-entry-heal: on
storage.fips-mode-rchecksum: on
transport.address-family: inet
nfs.disable: on
performance.client-io-threads: off

```

Ilustración 26 - Instalación de GlusterFS 12.

```

root@STR1-D11:~# gluster volume set ecommerce auth.allow 10.20.22.0/24
volume set: success

```

Ilustración 27 - Instalación de GlusterFS 13.

A partir de este momento está disponible el almacenamiento distribuido y redundado para montar en las máquinas en las que sea necesario.

4.3. PHP

El software *E-Commerce* Adobe Commerce Magento *Open source* se desarrolla con lenguaje PHP, concretamente en la versión instalada, depende de la versión mayor de PHP 8.1.

PHP es un lenguaje de programación basado en software libre, multiparadigma y débilmente tipado, de uso general, especialmente adaptado para el desarrollo web. Creado en 1994 por el programador danés Rasmus Lerdorf y cuyas siglas referenciaban inicialmente *Personal Home Page* pero que ha sido reemplazado por el inicialismo recursivo *Hypertext PreProcessor*.

El lenguaje es procesado por un servidor, ya sea un servidor web a través de un intérprete configurado como un módulo o demonio llamados por

CGI o directamente como un servicio residente alcanzable a través de un socket TCP/IP.

Esta capacidad de residir como un servicio independiente de CGI, independiente de un servidor web, y alcanzable mediante un socket, lo que permite distribuir y redundar muy fácilmente los *intérpretes* PHP que interpretarán el código, descargando la labor de responder a los clientes a los servidores web. Además, esta característica permite asignar en tiempo real qué *intérpretes* ejecutan qué procesos según su criticidad, pudiendo tener un pool de *intérpretes* de sólo lectura para las secciones públicas que no requieren transaccionalidad, y otros pools para secciones más críticas, que podrían contener medidas de seguridad mayores, creando así un servicio de interpretación de PHP por capas.

El programador de la comunidad Debian *Sury* mantiene una serie de repositorios habilitados de paquetes de PHP para Debian 10 y 11, entre los cuales se encuentra PHP8.1, versión necesaria para ejecutar Adobe Commerce Magento *Open source* 2.4.5.

El proceso de instalación comienza satisfaciendo una serie de dependencias de paquetes previas a añadir el repositorio a la lista de repositorios del gestor de paquetes de Debian:

```
root@BE1-D11:~# apt-get install ca-certificates apt-transport-https software-properties-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20210119).
The following additional packages will be installed:
  gir1.2-glib-2.0 gir1.2-packagekitglib-1.0 iso-codes libappstream4 libdw1 libgirepository-1.0-1 libglib2.0-0 libglib2.0-bin
  libglib2.0-data libgstreamer1.0-0 libpackagekit-glib2-18 libpolkit-agent-1-0 libpolkit-gobject-1-0 libstemmer0d libunwind8 libyaml-0-2
  lsb-release packagekit packagekit-tools policykit-1 python3-dbus python3-distro-info python3-gi python3-software-properties
  shared-mime-info unattended-upgrades xdg-user-dirs
Suggested packages:
  isoquery gstreamer1.0-tools appstream python-dbus-doc python3-dbus-dbg bsd-mailx needrestart powermgmt-base
The following NEW packages will be installed:
  apt-transport-https gir1.2-glib-2.0 gir1.2-packagekitglib-1.0 iso-codes libappstream4 libdw1 libgirepository-1.0-1 libglib2.0-0
  libglib2.0-bin libglib2.0-data libgstreamer1.0-0 libpackagekit-glib2-18 libpolkit-agent-1-0 libpolkit-gobject-1-0 libstemmer0d
  libunwind8 libyaml-0-2 lsb-release packagekit packagekit-tools policykit-1 python3-dbus python3-distro-info python3-gi
  python3-software-properties shared-mime-info software-properties-common unattended-upgrades xdg-user-dirs
0 upgraded, 29 newly installed, 0 to remove and 0 not upgraded.
Need to get 11.1 MB of archives.
After this operation, 55.0 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Ilustración 28 - Instalación de PHP 1.

```
root@BE1-D11:~# apt install gnupg2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  dirmngr gnupg gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm libassuan0 libksba8 libnpth0
  pinentry-curses
Suggested packages:
  dbus-user-session pinentry-gnome3 tor parcimonie xloadimage sdaemon pinentry-doc
The following NEW packages will be installed:
  dirmngr gnupg gnupg-l10n gnupg-utils gnupg2 gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm libassuan0 libksba8 libnpth0
  pinentry-curses
0 upgraded, 15 newly installed, 0 to remove and 0 not upgraded.
Need to get 8100 kB of archives.
After this operation, 16.1 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Ilustración 29 - Instalación de PHP 2.

A continuación, se añade la clave de cifrado del repositorio al anillo de claves del gestor de paquetes de Debian:


```

root@BE1-D11:~# wget -qO - https://packages.sury.org/php/apt.gpg | apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
root@BE1-D11:~#

```

Ilustración 30 - Instalación de PHP 3.

Finalmente se instalan los paquetes de PHP8.1 necesarios, está disponible en las páginas de conocimiento de Adobe, qué paquetes son necesarios para poder ejecutar el software *E-Commerce*:

```

root@BE1-D11:~# apt install php8.1 php8.1-pdo php8.1-mysqld php8.1-opcache php8.1-xml php8.1-gd php8.1-dev php8.1-mysql php8.1-intl php8.1-mbstring php8.1-bcmath php8.1-fpm php8.1-iconv php8.1-soap
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'php8.1-common' instead of 'php8.1-pdo'
Note, selecting 'php8.1-mysql' instead of 'php8.1-mysqld'
Note, selecting 'php8.1-common' instead of 'php8.1-iconv'
The following additional packages will be installed:
  autoconf automake autopoint autotools-dev binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp cpp-10 debhelper
  dh-autoreconf dh-strip-nondeterminism dpkg-dev dwz fakeroot fontconfig-config fonts-dejavu-core g++ g++-10 gcc gcc-10 gettext
  intltool-debian libabsl20200923 libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libaom0 libarchive-cpio-perl
  libarchive-zip-perl libasan6 libatomic1 libavif9 libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0
  libctf0 libdav1d4 libde265-0 libdebhelper-perl libdeflate0 libdpkg-perl libfakeroot libfile-fcntllock-perl
  libfile-stripnondeterminism-perl libfontconfig1 libfreetype6 libgav1-0 libgcc-10-dev libgd3 libgomp1 libheif1 libisl23 libitm1
  libjbig0 libjpeg62-turbo liblsan0 libltdl-dev libltdl7 libmail-sendmail-perl libmpc3 libmpfr6 libnsl-dev libnuma1 libonig5
  libpcre2-16-0 libpcre2-32-0 libpcre2-8-0 libpcre2-dev libpcre2-posix3 libpng16-16 libquadmath0 libsigsegv2 libsodium23 libssl-dev
  libstdc++-10-dev libsub-override-perl libsys-hostname-long-perl libtiff5 libtirpc-dev libtool libtsan0 libubsan1 libwebp6 libx11-6
  libx11-data libx265-192 libxau6 libxcb1 libxdmcp6 libxpm4 libxstl1.1 linux-libc-dev m4 make manpages-dev patch php-common php-pear
  php8.1-cli php8.1-readline pkg-config pkg-php-tools po-debconf shtool
Suggested packages:
  autoconf-archive gnu-standards autoconf-doc binutils-doc cpp-doc gcc-10-locales dh-make debian-keyring g++-multilib g++-10-multilib
  gcc-10-doc gcc-multilib flex bison gdb gcc-doc gcc-10-multilib gettext-doc libasprintf-dev libgettextpo-dev glibc-doc git bzr
  libgd-tools libtool-doc libssl-doc libstdc++-10-doc gfortran | fortran95-compiler gcj-jdk m4-doc make-doc ed diffutils-doc dh-php
  libmail-box-perl
The following NEW packages will be installed:
  autoconf automake autopoint autotools-dev binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp cpp-10 debhelper
  dh-autoreconf dh-strip-nondeterminism dpkg-dev dwz fakeroot fontconfig-config fonts-dejavu-core g++ g++-10 gcc gcc-10 gettext
  intltool-debian libabsl20200923 libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libaom0 libarchive-cpio-perl
  libarchive-zip-perl libasan6 libatomic1 libavif9 libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0
  libctf0 libdav1d4 libde265-0 libdebhelper-perl libdeflate0 libdpkg-perl libfakeroot libfile-fcntllock-perl
  libfile-stripnondeterminism-perl libfontconfig1 libfreetype6 libgav1-0 libgcc-10-dev libgd3 libgomp1 libheif1 libisl23 libitm1
  libjbig0 libjpeg62-turbo liblsan0 libltdl-dev libltdl7 libmail-sendmail-perl libmpc3 libmpfr6 libnsl-dev libnuma1 libonig5
  libpcre2-16-0 libpcre2-32-0 libpcre2-dev libpcre2-posix3 libpng16-16 libquadmath0 libsigsegv2 libsodium23 libssl-dev libstdc++-10-dev
  libsub-override-perl libsys-hostname-long-perl libtiff5 libtirpc-dev libtool libtsan0 libubsan1 libwebp6 libx11-6 libx11-data
  libx265-192 libxau6 libxcb1 libxdmcp6 libxpm4 libxstl1.1 linux-libc-dev m4 make manpages-dev patch php-common php-pear php8.1
  php8.1-bcmath php8.1-cli php8.1-common php8.1-dev php8.1-fpm php8.1-gd php8.1-intl php8.1-mbstring php8.1-mysql php8.1-opcache
  php8.1-readline php8.1-soap php8.1-xml pkg-config pkg-php-tools po-debconf shtool
The following packages will be upgraded:
  libpcre2-8-0
1 upgraded, 121 newly installed, 0 to remove and 2 not upgraded.
Need to get 89.2 MB of archives.
After this operation, 333 MB of additional disk space will be used.
Do you want to continue? [Y/n] |

```

Ilustración 31 - Instalación de PHP 4.

Es necesario ajustar ciertos parámetros en la configuración de PHP para que el servicio se comporte como es necesario según esta arquitectura, deberá poder atender las peticiones de los balanceadores en el puerto 9000, en la IP privada de LAN del contenedor, para ello se ha de editar el archivo de configuración del servicio PHP-FPM en `/etc/php/8.1/fpm/pool.d/www.conf` y modificar la sección de configuración:

```

; The address on which to accept FastCGI requests.
; Valid syntaxes are:
;   'ip.add.re.ss:port'    - to listen on a TCP socket to a specific IPv4 address on
;                           a specific port;
;   '[ip:6:addr:ess]:port' - to listen on a TCP socket to a specific IPv6 address on
;                           a specific port;
;   'port'                 - to listen on a TCP socket to all addresses
;                           (IPv6 and IPv4-mapped) on a specific port;
;   '/path/to/unix/socket' - to listen on a unix socket.
; Note: This value is mandatory.
;listen = /run/php/php8.1-fpm.sock
listen = 9000

```

Ilustración 32 - Instalación de PHP 5.

En este punto está disponible el servicio de interpretación de PHP en un container intérprete, este mismo container se clonará para crear un pool específico que atenderá las peticiones del *Backoffice*.

4.4. MySQL/MariaDB

MySQL y MariaDB son dos motores de bases de datos intercambiables entre sí, para ilustrar esta relación entre ambos SGBD, y por qué se selecciona como motor de bases de datos MaríaDB en detrimento de MySQL, observemos un poco de historia.

MySQL nace en MySQL AB, en 1995, empresa danesa fundada por tres desarrolladores, Michael Widenius, David Axmark and Allan Larsson. Creada con el objetivo de proveer un motor de bases de datos eficiente y confiable, tanto para el ámbito profesional como doméstico. Después de una decena de versiones alfa, sale a la luz en el año 2000, con licencia GPL, en sintonía con la comunidad *open source*, algo que trajo bastantes problemas a la empresa si bien consiguió salir adelante. Tan solo dos años después de su lanzamiento, MySQL ya había sido instalada millones de veces.

En el año 2008 Sun Microsystems adquiere MySQL AB, que hasta entonces era un reputado motor de BBDD de software libre, con un valor estimado de mil millones de dólares, si bien esta adquisición ya trajo serias dudas sobre cómo se iba a fraguar la nueva MySQL privada, la propuesta de Sun de proveer de un soporte dedicado no estuvo del todo mal visto. Sin embargo, al año siguiente Oracle Corporation compra a su vez Sun Microsystems. Estos movimientos corporativos sacuden la comunidad libre, de forma que Michalel Widenius, fundador de MySQL, tras el movimiento de compra por Sun, funda a su vez la Fundación MariaDB con el objetivo de mantener una bifurcación libre del motor MySQL, sin embargo tras la compra de Sun por parte de Oracle y entendiendo tal compra como un movimiento para reducir la competencia, surge el SGBD MariaDB, motor que posee las mismas órdenes, interfaces, bibliotecas y APIs de forma que se puede cambiar un servidor por otro directamente.

El software *E-Commerce* Adobe Commerce Magento *Open source*, tiene como requisito para la gestión de la base de datos a los SGBD MariaDB en versión 10.4 o MySQL en versión 8.0. En este caso se ha instalado MariaDB 10.4 como SGBD sobre Debian 10, evitando pues tener que añadir repositorios de terceros, manteniendo la línea de licenciamiento GPL, y permitiendo la gestión de actualizaciones de seguridad directamente desde los repositorios de la distribución.

La comunidad de MariaDB mantiene una serie de repositorios para distribuciones y versiones específicas, en este caso añadiremos los repositorios de MariaDB 10.4 para Debian 10. La versión MariaDB 10.4 no se encuentra disponible en Debian 11, y tampoco se mantienen repositorios que permitan instalar esta versión del SGBD en la distribución, siendo por defecto la versión MariaDB 10.5 y únicamente pudiendo instalar versiones superiores a esta sin tener que compilar manualmente la aplicación.

Se siguen los pasos necesarios obtenidos de la página de documentación del sitio oficial de MariaDB para añadir los repositorios de la versión 10.4 en Debian 10:

```

root@sdbd1:~# apt install apt-transport-https curl gnupg
Reading package lists... Done
Building dependency tree... Done
The following additional packages will be installed:
  dirmngr gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm libassuan0 libcurl4 libksba8 libnpth0
  pinentry-curses
Suggested packages:
  dbus-user-session pinentry-gnome3 tor parcimonie xloadimage sddm pinentry-doc
The following NEW packages will be installed:
  apt-transport-https curl dirmngr gnupg gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm libassuan0
  libcurl4 libksba8 libnpth0 pinentry-curses
0 upgraded, 17 newly installed, 0 to remove and 0 not upgraded.
Need to get 7834 kB of archives.
After this operation, 16.2 MB of additional disk space will be used.
Do you want to continue? [Y/n]

```

Ilustración 33 - Instalación de MariaDB 1.

```

root@sdbd1:~# curl -o /etc/apt/trusted.gpg.d/mariadb_release_signing_key.asc 'https://mariadb.org/mariadb_release_signing_key.asc'
% Total % Received % Xferd Average Speed Time Time Time Current
          Dload Upload Total Spent Left Speed
100 6575 100 6575 0 0 8260 0 --:--:-- --:--:-- --:--:-- 8249
root@sdbd1:~# sh -c "echo 'deb https://mirrors.up.pt/pub/mariadb/repo/10.4/debian buster main' >>/etc/apt/sources.list"
root@sdbd1:~#

```

Ilustración 34 - Instalación de MariaDB 2.

Se instala el SGBD:

```

root@SGBD1-D10:~# apt install mariadb-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  galera-4 gawk libaio1 libcgf-fast-perl libcgf-pm-perl libdbd-mysql-perl libdbi-perl libencode-locale-perl libfcgi-perl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libmariadb3 libreadline5 libsigsegv2 libterm-readkey-perl libtimedate-perl liburi-perl mariadb-client-10.4
  mariadb-client-core-10.4 mariadb-common mariadb-server-10.4 mariadb-server-core-10.4 mysql-common psmisc rsync socat
Suggested packages:
  gawk-doc libclone-perl libmldbm-perl libnet-daemon-perl libsql-statement-perl libdata-dump-perl libipc-sharedcache-perl libwww-perl
  mailx mariadb-test netcat-openbsd tinycu
The following NEW packages will be installed:
  galera-4 gawk libaio1 libcgf-fast-perl libcgf-pm-perl libdbd-mysql-perl libdbi-perl libencode-locale-perl libfcgi-perl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libmariadb3 libreadline5 libsigsegv2 libterm-readkey-perl libtimedate-perl liburi-perl mariadb-client-10.4
  mariadb-client-core-10.4 mariadb-common mariadb-server mariadb-server-10.4 mariadb-server-core-10.4 mysql-common psmisc rsync socat
0 upgraded, 32 newly installed, 0 to remove and 0 not upgraded.
Need to get 28.9 MB of archives.
After this operation, 219 MB of additional disk space will be used.
Do you want to continue? [Y/n]

```

Ilustración 35 - Instalación de MariaDB 3.

A continuación se habilita y arranca el servicio de SGBD:

```

root@SGBD1-D11:~# systemctl enable mariadb
Synchronizing state of mariadb.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable mariadb
root@SGBD1-D11:~# systemctl start mysql
root@SGBD1-D11:~#

```

Ilustración 36 - Instalación de MariaDB 4.

En instalaciones nuevas del SGBD MariaDB o MySQL es importante ejecutar el script incluido en la instalación “mysql_secure_installation” el cual solicitará contraseña para el usuario root del SGBD (no confundir con el usuario root del sistema) y permitirá cambiar los métodos y orígenes de autenticación de usuarios, así como eliminar las bases de datos de pruebas que vienen incluidas en la instalación.

```
root@SGBD1-D11:~# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] n
... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] n
... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
root@SGBD1-D11:~#
```

Ilustración 37 - Instalación de MariaDB 5.

Una vez el SGBD está securizado, se crean la base de datos y el usuario correspondiente con sus privilegios para el software Adobe Commerce Magento *Open source*:

```

root@SGBD1-D11:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 37
Server version: 10.5.15-MariaDB-0+deb11u1 Debian 11

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 ecommerce;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> CREATE USER 'ecomm_user'@'%' IDENTIFIED BY 'VJXsRmwRDGeHsT3m';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON ecommerce.* TO 'ecomm_user'@'%';
Query OK, 0 rows affected (0.005 sec)

```

Ilustración 38 - Instalación de MariaDB 6.

Finalmente se recargan las tablas de permisos:

```

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]>

```

Ilustración 39 - Instalación de MariaDB 7.

Se configura el servicio MySQL para que escuche en la IP y puertos necesarios para la aplicación, para ello se edita el archivo de configuración por defecto de MariaDB (/etc/mysql/my.cnf) con la siguiente configuración:

```

# This will be passed to all mysql clients
# It has been reported that passwords should be enclosed with ticks/quotes
# especially if they contain "#" chars...
# Remember to edit /etc/mysql/debian.cnf when changing the socket location.
[client]
port                = 3306
socket              = /var/run/mysqld/mysqld.sock

```

Ilustración 40 - Instalación de MariaDB 8.

```

#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address        = 0.0.0.0

```

Ilustración 41 - Instalación de MariaDB 9.

Alcanzado este punto, el SGBD y la base de datos necesaria para el funcionamiento de Adobe Commerce Magento *Open source* están preparadas para trabajar.

4.5. Redis

Un componente importante, aunque no estrictamente necesario para el funcionamiento del software Adobe Commerce Magento *Open source* es la base de datos Redis.

Redis es una base de datos en memoria que usa el tipo de almacenamiento de tablas de hash, o clave/valor, aunque puede ser usada como base de datos persistente. Escrita en lenguaje ANSI C por el programador Salvatore Sanfilippo y licenciada bajo el estándar BSD, por lo que se la considera software de código abierto.

Mediante el uso de este SGBD, se pretende conseguir agilidad y velocidad extras del aplicativo, permitiendo configurar la caché de sesiones, la caché de página y la de *backend* de Magento directamente en un servicio en memoria, evitando así la lectura escritura a disco y aumentando dramáticamente la velocidad y la respuesta del sitio web.

El requisito de versión para Adobe Commerce Magento *Open source*, es Redis 6.2 o superior. Como los paquetes en el repositorio oficial de Debian tienen como estable la versión de Redis 6.0, insuficiente para satisfacer este requisito, se agregan los repositorios oficiales de Redis para Debian, lo cual nos permitirá instalar la versión 7.0, la cual si satisface la dependencia de versión al ser superior a 6.2.

Se instalan las dependencias previas para la agregar el repositorio de Redis a Debian 11:

```
root@RDS1-D11:~# apt install wget curl gnupg
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wget is already the newest version (1.21-1+deb11u1).
The following packages were automatically installed and are no longer required:
 libatomic1 libjemalloc2 liblua5.1-0 liblzfl lua-bitop lua-cjson
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
 dirnmgr gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm libassuan0 libcurl4
 libksba8 libnpt0 pinentry-curses
Suggested packages:
 dbus-user-session pinentry-gnome3 tor parcimonie xloadimage sddm pinentry-doc
The following NEW packages will be installed:
 curl dirnmgr gnupg gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm libassuan0
 libcurl4 libksba8 libnpt0 pinentry-curses
0 upgraded, 16 newly installed, 0 to remove and 0 not upgraded.
Need to get 8280 kB of archives.
After this operation, 16.9 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Ilustración 42 - Instalación de Redis 1.

```
root@BE1-D11:~# apt-get install ca-certificates apt-transport-https software-properties-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20210119).
The following additional packages will be installed:
 gir1.2-glib-2.0 gir1.2-packagekitglib-1.0 iso-codes libappstream4 libdw1 libgirepository-1.0-1 libglib2.0-0 libglib2.0-bin
 libglib2.0-data libgstreamer1.0-0 libpackagekit-glib2-18 libpolkit-agent-1-0 libpolkit-gobject-1-0 libstemmer0d libunwind8 libyam1-0-2
 lsb-release packagekit packagekit-tools policykit-1 python3-dbus python3-distro-info python3-gi python3-software-properties
 shared-mime-info unattended-upgrades xdg-user-dirs
Suggested packages:
 isoquery gstreamer1.0-tools appstream python-dbus-doc python3-dbus-dbg bsd-mailx needrestart powermgmt-base
The following NEW packages will be installed:
 apt-transport-https gir1.2-glib-2.0 gir1.2-packagekitglib-1.0 iso-codes libappstream4 libdw1 libgirepository-1.0-1 libglib2.0-0
 libglib2.0-bin libglib2.0-data libgstreamer1.0-0 libpackagekit-glib2-18 libpolkit-agent-1-0 libpolkit-gobject-1-0 libstemmer0d
 libunwind8 libyam1-0-2 lsb-release packagekit packagekit-tools policykit-1 python3-dbus python3-distro-info python3-gi
 python3-software-properties shared-mime-info software-properties-common unattended-upgrades xdg-user-dirs
0 upgraded, 29 newly installed, 0 to remove and 0 not upgraded.
Need to get 11.1 MB of archives.
After this operation, 55.0 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Ilustración 43 - Instalación de Redis 2.

Se añade la clave de cifrado de los paquetes del repositorio y la configuración del repositorio al conjunto de fuentes del gestor de paquetes de Debian 11:

```
root@RDS1-D11:~# curl https://packages.redis.io/gpg | apt-key add -
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
ted.gpg.d instead (see apt-key(8)).
100 3155 100 3155 0 0 1820 0 0:00:01 0:00:01 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
OK
```

Ilustración 44 - Instalación de Redis 3.

```

root@RDS1-D11:~# echo "deb https://packages.redis.io/deb $(lsb_release -cs) main" | tee /etc/apt/sources.list.d/redis.list
deb https://packages.redis.io/deb bullseye main
root@RDS1-D11:~# |

```

Ilustración 45 - Instalación de Redis 4.

Se actualiza el listado de paquetes e instala el software Redis:

```

root@RDS1-D11:~# apt update
Get:1 http://security.debian.org bullseye-security InRelease [48.4 kB]
Hit:2 http://ftp.debian.org/debian bullseye InRelease
Get:3 http://ftp.debian.org/debian bullseye-updates InRelease [44.1 kB]
Get:4 http://security.debian.org bullseye-security/main amd64 Packages [203 kB]
Get:5 http://security.debian.org bullseye-security/main Translation-en [132 kB]
Get:6 https://packages.redis.io/deb bullseye InRelease [3826 B]
Get:7 https://packages.redis.io/deb bullseye/main amd64 Packages [7493 B]
Get:8 https://packages.redis.io/deb bullseye/main all Packages [1604 B]
Fetched 440 kB in 2s (242 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
5 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@RDS1-D11:~# apt install redis-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libatomic1 libjemalloc2 liblua5.1-0 liblzfl1 lua-bitop lua-cjson
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  redis-tools
Suggested packages:
  ruby-redis
The following NEW packages will be installed:
  redis-server redis-tools
0 upgraded, 2 newly installed, 0 to remove and 5 not upgraded.
Need to get 1330 kB of archives.
After this operation, 8561 kB of additional disk space will be used.
Do you want to continue? [Y/n] |

```

Ilustración 46 - Instalación de Redis 5.

Para el correcto funcionamiento distribuido, y con el fin de que Redis permita escribir datos en un entorno de red, es necesario modificar dos parámetros en el archivo de configuración del servicio (/etc/redis/redis.conf), deshabilitando el modo protegido y permitiendo al servicio “escuchar” en todas las direcciones IP:

```

# By default, outgoing connections (from replica to master, from Sentinel to
# instances, cluster bus, etc.) are not bound to a specific local address. In
# most cases, this means the operating system will handle that based on routing
# and the interface through which the connection goes out.
#
# Using bind-source-addr it is possible to configure a specific address to bind
# to, which may also affect how the connection gets routed.
#
# Example:
# bind-source-addr 10.0.0.1

```

Ilustración 47 - Instalación de Redis 6.

```

# By default protected mode is enabled. You should disable it only if
# you are sure you want clients from other hosts to connect to Redis
# even if no authentication is configured.
protected-mode no

```

Ilustración 48 - Instalación de Redis 7.

Finalmente se arranca y habilita el servicio:

```

root@RDS1-D11:~# systemctl enable redis-server
Synchronizing state of redis-server.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable redis-server
root@RDS1-D11:~# systemctl start redis-server
root@RDS1-D11:~# |

```

Ilustración 49 - Instalación de Redis 8.

Por lo que el servicio de Redis queda disponible.

4.6. Elasticsearch

Uno de los principales problemas de lentitud en un *E-Commerce* moderno, con potencialmente decenas de miles de productos y cientos de categorías son las búsquedas. Originalmente el software de *E-Commerce* Adobe Commerce Magento *Open source* utilizaba llamadas estándar a la base de datos para encontrar los artículos o categorías solicitadas por el usuario en el formulario de búsqueda, esta solución es extremadamente lenta y eficiente, por lo que las versiones modernas del aplicativo soportan dos servicios de búsqueda auto alojables, tales como Elasticsearch u Opensearch. En este caso se ha utilizado el software Elasticsearch por ser más maduro y con mayor acceso a documentación.

Elasticsearch es un servicio de búsqueda basado en la API Lucene, específica para recuperación de información, que provee de un motor de búsqueda completo con interfaz Restful con documentos basados en JSON, desarrollado en Java. Ambos softwares, Lucene y Elasticsearch son de código abierto bajo las condiciones de licencia Apache.

Para instalar la versión necesaria de Elasticsearch hay una guía específica tanto para la versión de Debian como para la versión necesaria para Magento 2.4.5, la cual se seguirá paso a paso.

Se instalan los paquetes necesarios para poder añadir el repositorio del software al administrador de paquetes de Debian11:

```
root@ELS1-D11:~# apt install gnupg apt-transport-https
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  dirmngr gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm libassuan0 libksba8 libnpt0 pinentry-curses
Suggested packages:
  dbus-user-session pinentry-gnome3 tor parcimonie xloadimage scd daemon pinentry-doc
The following NEW packages will be installed:
  apt-transport-https dirmngr gnupg gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm libassuan0 libksba8
  libnpt0 pinentry-curses
0 upgraded, 15 newly installed, 0 to remove and 0 not upgraded.
Need to get 7825 kB of archives.
After this operation, 15.8 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Ilustración 50 - Instalación de Elasticsearch 1.

A continuación se añade la clave gpg que cifra todos los paquetes del repositorio de Elasticsearch:

```
root@ELS1-D11:~# wget -qO - https://artifacts.elastic.co/gpg-KEY-elasticsearch | gpg --dearmor -o /usr/share/keyrings/elasticsearch-keyring
g.gpg
root@ELS1-D11:~#
```

Ilustración 51 - Instalación de Elasticsearch 2.

Finalmente el se agrega el repositorio al administrador de paquetes:

```
root@ELS1-D11:~# echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/7.x/apt stable
main" | tee /etc/apt/sources.list.d/elastic-7.x.list
deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/7.x/apt stable main
root@ELS1-D11:~#
```

Ilustración 52 - Instalación de Elasticsearch 3.

Se instalan los paquetes y establece el arranque del servicio:


```
root@ELS1-D11:~# apt install elasticsearch
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
 elasticsearch
0 upgraded, 1 newly installed, 0 to remove and 6 not upgraded.
Need to get 315 MB of archives.
After this operation, 526 MB of additional disk space will be used.
```

Ilustración 53 - Instalación de Elasticsearch 4.

```
root@ELS1-D11:~# systemctl daemon-reload
root@ELS1-D11:~# systemctl enable elasticsearch.service
Synchronizing state of elasticsearch.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable elasticsearch
Created symlink /etc/systemd/system/multi-user.target.wants/elasticsearch.service -> /lib/systemd/system/elasticsearch.service.
```

Ilustración 54 - Instalación de Elasticsearch 5.

```
root@ELS1-D11:~# systemctl start elasticsearch.service
root@ELS1-D11:~# |
```

Ilustración 55 - Instalación de Elasticsearch 6.

Es importante configurar el servicio para que use la cantidad de memoria correspondiente al contenedor, en este caso se ha configurado con 4Gb de RAM, pero debido a cómo se presentan los recursos de sistema a un contenedor, la máquina virtual Java sobre la que se ejecuta Elasticsearch “ve” toda la memoria disponible en el servidor, intentando arrancar así hasta 32Gb de RAM, lo cual provoca un OOM por parte del sistema operativo y un fallo en el arranque del servicio.

Para realizar esta configuración, simplemente se edita el archivo de opciones de la JVM en /etc/Elasticsearch/jvm.options y se modifican los parámetros Xms y Xmx, siendo la cantidad reservada inicial Xms y la cantidad Xmx, una configuración coherente con las características del contenedor será:

```
#####
## IMPORTANT: JVM heap size
#####
##
## The heap size is automatically configured by Elasticsearch
## based on the available memory in your system and the roles
## each node is configured to fulfill. If specifying heap is
## required, it should be done through a file in jvm.options.d,
## and the min and max should be set to the same value. For
## example, to set the heap to 4 GB, create a new file in the
## jvm.options.d directory containing these lines:
##
-Xms2g
-Xmx3g
```

Ilustración 56 - Instalación de Elasticsearch 7.

Alcanzado este hito estará disponible el servicio Elasticsearch para su uso en Adobe Commerce Magento *Open source*.

4.7. Varnish y Nginx

Estos dos softwares van en la misma sección debido a que se instalan en tándem en el mismo tipo de servidor: el balanceador.

Nginx es un servidor Web ligero de alto rendimiento, libre y de código abierto bajo licencia BSD, se ha seleccionado en detrimento de Apache por experiencia propia administrando servidores web basados en esta tecnología.

Varnish es un sistema de full page caché, un acelerador HTTP diseñado para soportar gran carga de peticiones, utiliza la memoria virtual del sistema para cachear los objetos solicitados en las peticiones de los clientes, las cuales

se satisfacen mediante un pool de hilos separados, que a su vez se alimentan de una cola común de peticiones. Estos objetos pueden ser de cualquier tipo, contenido, scripts, páginas pre compiladas etc. Esta flexibilidad permite acelerar la velocidad de un sitio en orden de varias centenas porcentuales, si bien tiene el gran inconveniente de que debe haber una configuración adecuada de qué se debe y qué no se debe cachear, además de un sistema de purgado rápido y eficiente. Magento soporta full page caché con Varnish en su propio núcleo, lo cual hace que esta configuración sea extremadamente sencilla en comparación con otras aplicaciones web.

La instalación de Varnish es muy sencilla en este contexto, dado que la configuración la obtendremos de Magento y las versiones del repositorio de paquetes de la distribución Debian contienen una versión compatible y adecuada, por lo que se instala el paquete del repositorio:

```
root@LB1-D11:~# apt install varnish
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
 binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-10 gcc gcc-10 libasan6 libatomic1 libbinutils libc-dev-bin libc-devtools
 libc6-dev libccl-0 libcrypt-dev libctf-nobfd0 libctf0 libgcc-10-dev libgomp1 libisl23 libitm1 libjemalloc2 liblsan0 libmpc3 libmpfr6
 libnsl-dev libquadmath0 libtirpc-dev libtsan0 libubsan1 libvarnishapi2 linux-libc-dev manpages-dev
Suggested packages:
 binutils-doc cpp-doc gcc-10-locales gcc-multilib make autoconf automake libtool flex bison gdb gcc-doc gcc-10-multilib gcc-10-doc
 glibc-doc varnish-doc
The following NEW packages will be installed:
 binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-10 gcc gcc-10 libasan6 libatomic1 libbinutils libc-dev-bin libc-devtools
 libc6-dev libccl-0 libcrypt-dev libctf-nobfd0 libctf0 libgcc-10-dev libgomp1 libisl23 libitm1 libjemalloc2 liblsan0 libmpc3 libmpfr6
 libnsl-dev libquadmath0 libtirpc-dev libtsan0 libubsan1 libvarnishapi2 linux-libc-dev manpages-dev varnish
0 upgraded, 34 newly installed, 0 to remove and 0 not upgraded.
Need to get 49.9 MB of archives.
After this operation, 182 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Ilustración 57 - Instalación de Varnish 1.

Y se habilita el servicio:

```
root@LB1-D11:~# systemctl enable varnish
Synchronizing state of varnish.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable varnish
root@LB1-D11:~# systemctl start varnish
```

Ilustración 58 - Instalación de Varnish 2.

En el siguiente punto de este capítulo se revisará la configuración de Varnish, así como la obtención del archivo de configuración necesario para el correcto funcionamiento del caché de página.

Juntamente con Varnish, se instala Nginx cuya instalación es también directa, se instala directamente de los repositorios de la distribución:

```

root@LB1-D11:/media/ecommerce# apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
 fontconfig-config fonts-dejavu-core geop-database libdeflate0 libfontconfig1 libfreetype6 libgd3 libgeoip1 libjbig0 libjpeg62-turbo
 libnginx-mod-http-geoip libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
 libnginx-mod-stream-geoip libpng16-16 libtiff5 libwebp6 libx11-6 libx11-data libxau6 libxcb1 libxdmcp6 libxpm4 libxslt1.1 nginx-common
 nginx-core
Suggested packages:
 libgd-tools geop-bin fcgiwrap nginx-doc
The following NEW packages will be installed:
 fontconfig-config fonts-dejavu-core geop-database libdeflate0 libfontconfig1 libfreetype6 libgd3 libgeoip1 libjbig0 libjpeg62-turbo
 libnginx-mod-http-geoip libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
 libnginx-mod-stream-geoip libpng16-16 libtiff5 libwebp6 libx11-6 libx11-data libxau6 libxcb1 libxdmcp6 libxpm4 libxslt1.1 nginx
 nginx-common nginx-core
0 upgraded, 29 newly installed, 0 to remove and 0 not upgraded.
Need to get 9427 kB of archives.
After this operation, 25.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] |

```

Ilustración 59 - Instalación de Nginx 1.

Y posteriormente se habilita el servicio:

```

root@LB1-D11:/etc/nginx/sites-available# systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable nginx
root@LB1-D11:/etc/nginx/sites-available# systemctl start nginx

```

Ilustración 60 - Instalación de Nginx 2.

La configuración del balanceador consiste en una primera etapa de proxy reverso y *offloading* de SSL por parte de Nginx, que pasará la petición a Varnish si esta ha de ser cacheada, o de nuevo a otra instancia de Nginx escuchando en un puerto diferente, que a su vez actúa como balanceador de conexiones TCP a los *pools* de *intérpretes* de PHP tanto para *frontend* como para *backoffice*. Las peticiones al *backoffice*, por defecto nunca serán cacheadas, esto resta agilidad a la gestión de este, pero ofrece la total seguridad de que ningún dato crítico está cacheado.

El flujo de información es el siguiente:

- Se recibe una petición de cliente por protocolo HTTP (80), se reescribe automáticamente a HTTPS (443).
- Se recibe una petición de cliente por protocolo HTTPS (443), si esta es de *frontend* se envía mediante proxy a Varnish (6081)
- Se recibe una petición de cliente por protocolo HTTPS (443), si esta es de *backend* se envía mediante proxy a Nginx (8080)
- Se recibe una petición de Nginx a Varnish, si esta está cacheada previamente se devuelve, en caso contrario se envía mediante proxy a Nginx (8080)
- Se recibe una petición de Varnish o Nginx a Nginx (8080) se hace proxy reverso de la petición al pool PHP correspondiente.

Esta configuración permite tener tantos balanceadores como IP externas sea preciso tener para dar servicio a la plataforma, ya sea con objetivo de alta disponibilidad y tolerancia a fallos, como de rendimiento o ambos objetivos a la vez.

Un gráfico ilustrativo de este flujo de trabajo es el que se presenta en la figura a continuación:

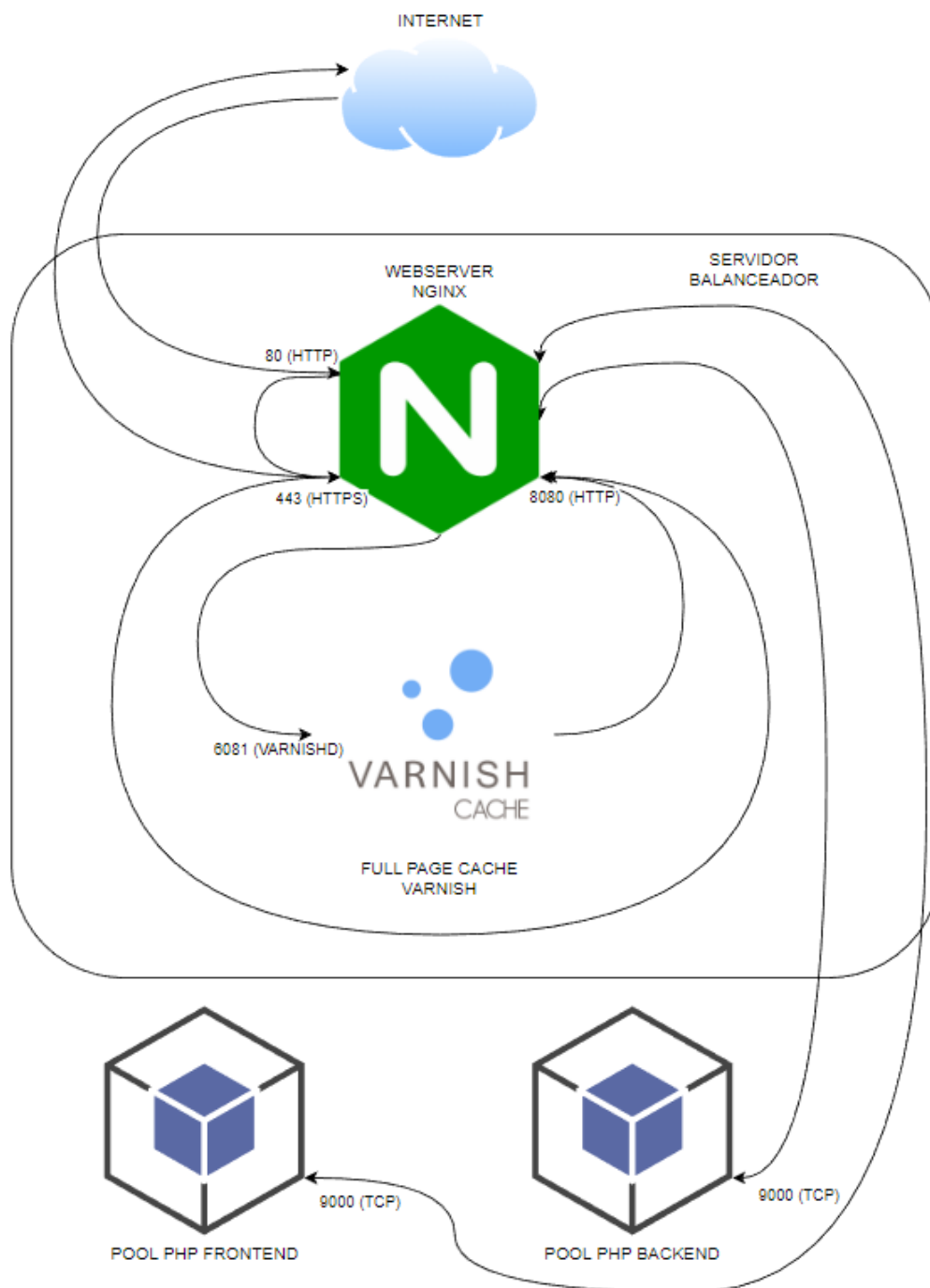


Ilustración 61 - Flujo de entrada de peticiones de cliente.

Para poder ofrecer esta funcionalidad, se configuran dos hosts virtuales en Nginx, el externo que atenderá las peticiones desde internet en los puertos 80 y 443, encargándose del cifrado SSL y de reescribir peticiones no seguras, para a continuación enviar por proxy reverso a Varnish o al host virtual interno. Por otro lado, el host virtual interno que atenderá las peticiones en el puerto 8080 y hará de proxy reverso hacia los intérpretes PHP.

4.8. Adobe Commerce Magento *Open source*

Todos los componentes software han sido instalados en sus respectivos servidores, por lo que es posible avanzar al paso final en cuanto a software se refiere.

Un requisito previo para poder hacer la instalación de Adobe Commerce Magento *Open source* es tener una cuenta en el sitio de Adobe, la cual nos proporcionará un usuario y contraseña numéricos necesarios para poder acceder al repositorio y descargar el software.

El portal de conocimiento de Magento tiene diversos tutoriales para realizar la instalación del aplicativo, en este trabajo se ha utilizado el software composer para realizar la instalación de este:

```
www-data@control:~$ composer create-project --repository-url=https://repo.magento.com/ magento/project-community-edition /media/ecommerce/
current
PHP Deprecated: Return type of Symfony\Component\Console\Helper\HelperSet::getIterator() should either be compatible with IteratorAggregate:
getIterator(): Traversable, or the #[\ReturnTypeWillChange] attribute should be used to temporarily suppress the notice in /usr/share/
php/Symfony/Component/Console/Helper/HelperSet.php on line 103

Deprecated: Return type of Symfony\Component\Console\Helper\HelperSet::getIterator() should either be compatible with IteratorAggregate::g
etIterator(): Traversable, or the #[\ReturnTypeWillChange] attribute should be used to temporarily suppress the notice in /usr/share/php/S
ymfony/Component/Console/Helper/HelperSet.php on line 103
Creating a "magento/project-community-edition" project at "/media/ecommerce/current"
Authentication required (repo.magento.com):
  Username: 8c5df1bb071b59610f102e0b5c355fd
  Password:
Do you want to store credentials for repo.magento.com in /var/www/.config/composer/auth.json ? [Yn] y
Installing magento/project-community-edition (2.4.5-p1)
- Downloading magento/project-community-edition (2.4.5-p1)
- Installing magento/project-community-edition (2.4.5-p1): Extracting archive
Created project in /media/ecommerce/current
Loading composer repositories with package information
Info from https://repo.packagist.org: #StandWithUkraine
```

Ilustración 62 - Instalación de Adobe Commerce Magento *Open source* 1.

El instalador genera gran cantidad de salida por pantalla, la mayoría de la cual son mensajes de advertencia los cuales no influyen en el resultado final de la propia instalación, si bien no se pueden desactivar, por lo que se ha truncado y omitido gran parte de dicha salida. El proceso de instalación descargará y descomprimirá el software magento y todas sus dependencias usando composer. Pasados unos minutos el software se habrá descargado y descomprimido en la ruta seleccionada y se podrá proceder a la instalación, para la que se deben tener a mano los parámetros del servicio de Base de Datos así como el servicio de Elasticsearch, el resto de servicios, como Redis o Varnish se configuran una vez ya se ha instalado Adobe Commerce Magento *Open source*.

Se instala Adobe Commerce Magento *Open source* con el comando siguiente:

```
www-data@control:/media/ecommerce/current$ bin/magento setup:install \
--base-url=http://magento.ulla-yanez.local \
--db-host=10.20.22.40 \
--db-ecommerce \
--db-user=ecomm_user \
--db-password=VJXsRmWRDGeHsT3m \
--admin-firstname=admin \
--admin-lastname=admin \
--admin-email=jose@ulla-yanez.com \
--admin-user=admin \
--admin-password=5T5HeXtGmfHx8cvy \
--language=es_ES \
--currency=EUR \
--timezone=Europe/Madrid \
--use-rewrites=1 \
--search-engine=elasticsearch7 \
--elasticsearch-host=10.20.22.60 \
--elasticsearch-port=9200 \
--elasticsearch-index-prefix=magento2 \
--elasticsearch-timeout=15
```

Ilustración 63 - Instalación de Adobe Commerce Magento *Open source* 2.

Al igual que en la descarga, se omite la salida de la instalación, cuando esta termine nos mostrará un mensaje de éxito y la URI de acceso al Backoffice:

```
[SUCCESS]: Magento installation complete.  
[SUCCESS]: Magento Admin URI: /admin_1qw7wb  
Nothing to import.
```

Ilustración 64 - Instalación de Adobe Commerce Magento *Open source* 3.

A continuación, se configuran las distintas cachés que se guardarán en el servidor Redis, para ello se utilizarán 3 bases de datos distintas dentro del mismo servidor Redis, como por defecto Redis tiene preconfiguradas 16 bases de datos distintas (0..15), se utilizarán las 3 primeras, de forma que:

Backend, Base de Datos 0:

```
www-data@control:/media/ecommerce/current$ bin/magento setup:config:set --cache-backend=redis --cache-backend-redis-server=10.20.22.50 --c  
ache-backend-redis-port=6379 --cache-backend-redis-db=0  
We saved default values for these options: remote-storage-prefix, remote-storage-key, remote-storage-secret, remote-storage-path-style, am  
qp-host, amqp-port, amqp-user, amqp-password, amqp-virtualhost, amqp-ssl, amqp-ssl-options, db-ssl-key, db-ssl-cert, db-ssl-ca, db-ssl-ver  
ify, allow-parallel-generation.
```

Ilustración 65 - Instalación de Adobe Commerce Magento *Open source* 4.

Página, Base de Datos 1:

```
www-data@control:/media/ecommerce/current$ bin/magento setup:config:set --page-cache=redis --page-cache-redis-server=10.20.22.50 --page-ca  
che-redis-db=1  
We saved default values for these options: remote-storage-prefix, remote-storage-key, remote-storage-secret, remote-storage-path-style, am  
qp-host, amqp-port, amqp-user, amqp-password, amqp-virtualhost, amqp-ssl, amqp-ssl-options, db-ssl-key, db-ssl-cert, db-ssl-ca, db-ssl-ver  
ify, allow-parallel-generation.
```

Ilustración 66 - Instalación de Adobe Commerce Magento *Open source* 5.

Y Sesión, Base de datos 3:

```
www-data@control:/media/ecommerce/current$ bin/magento setup:config:set --session-save=redis --session-save-redis-host=10.20.22.50 --sessi  
on-save-redis-log-level=4 --session-save-redis-db=2  
Overwrite the existing configuration for session-save?[Y/n]Y  
Overwrite the existing configuration for allow-parallel-generation?[Y/n]n  
We saved default values for these options: remote-storage-prefix, remote-storage-key, remote-storage-secret, remote-storage-path-style, am  
qp-host, amqp-port, amqp-user, amqp-password, amqp-virtualhost, amqp-ssl, amqp-ssl-options, db-ssl-key, db-ssl-cert, db-ssl-ca, db-ssl-ver  
ify.
```

Ilustración 67 - Instalación de Adobe Commerce Magento *Open source* 6.

No estrictamente necesario y por precaución, se vuelve a habilitar la ejecución paralela del procesamiento de caché:

```
www-data@control:/media/ecommerce/current$ bin/magento setup:config:set --allow-parallel-generation  
Overwrite the existing configuration for allow-parallel-generation?[Y/n]Y  
We saved default values for these options: remote-storage-prefix, remote-storage-key, remote-storage-secret, remote-storage-path-style, am  
qp-host, amqp-port, amqp-user, amqp-password, amqp-virtualhost, amqp-ssl, amqp-ssl-options, db-ssl-key, db-ssl-cert, db-ssl-ca, db-ssl-ver  
ify.
```

Ilustración 68 - Instalación de Adobe Commerce Magento *Open source* 7.

Finalmente, por conveniencia, se carga un set de datos preconfigurado de Magento, mediante esta carga se tendrán datos ficticios de productos, categorías, usuarios, administradores y compras:

```
www-data@control:/media/ecommerce/current$ bin/magento setup:perf:generate-fixtures ./setup/performance-toolkit/profiles/ce/small.xml
```

Ilustración 69 - Instalación de Adobe Commerce Magento *Open source* 8.

El resto de la configuración de Adobe Commerce Magento *Open source* se hará a través del *backoffice*, por lo que se procede a cargar desde un navegador usando la URL https://magento.ulla-yanez.local/admin_1qw7wb:

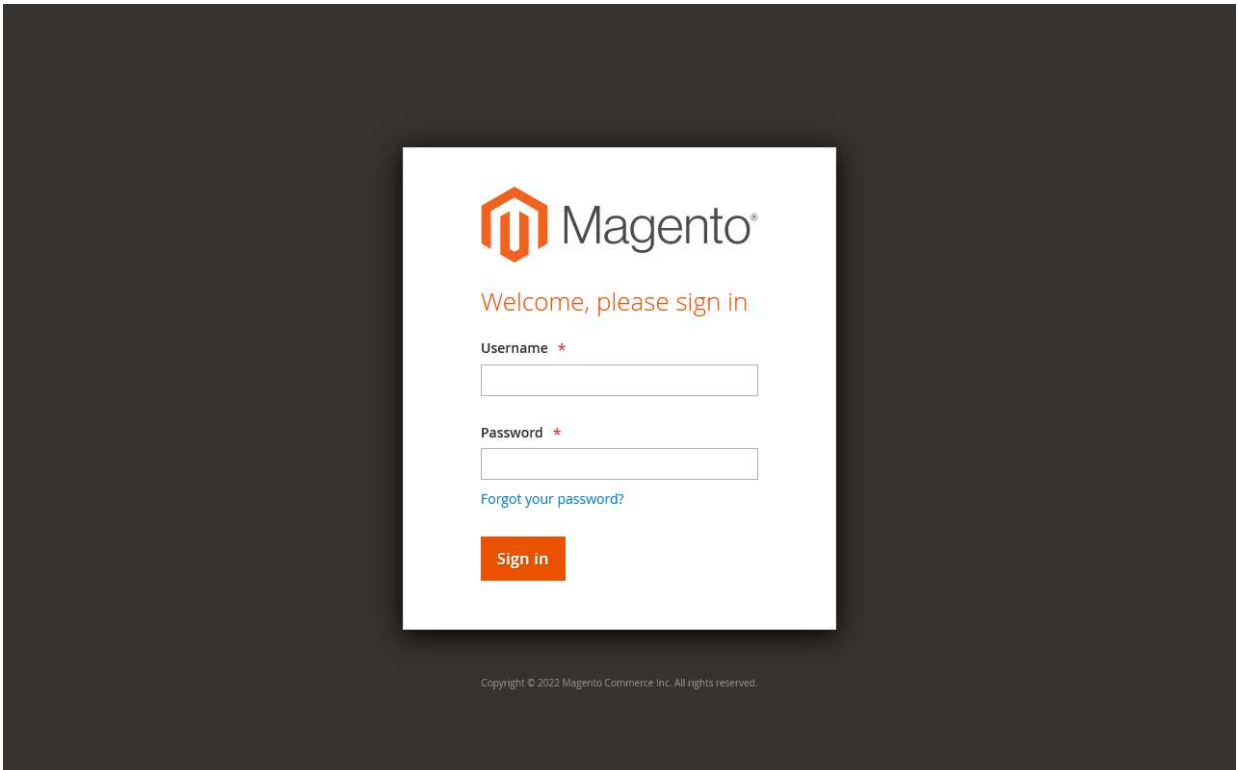


Ilustración 70 - Adobe Commerce Magento *Open source*. Login Backoffice.

Para activar la “Full Page Cache” usando Varnish, en la configuración de Backoffice, se accede a Stores -> Configuration -> Advanced -> System, en la sección Full Page Cache se configura con los parámetros generados en el capítulo de instalación de Varnish:

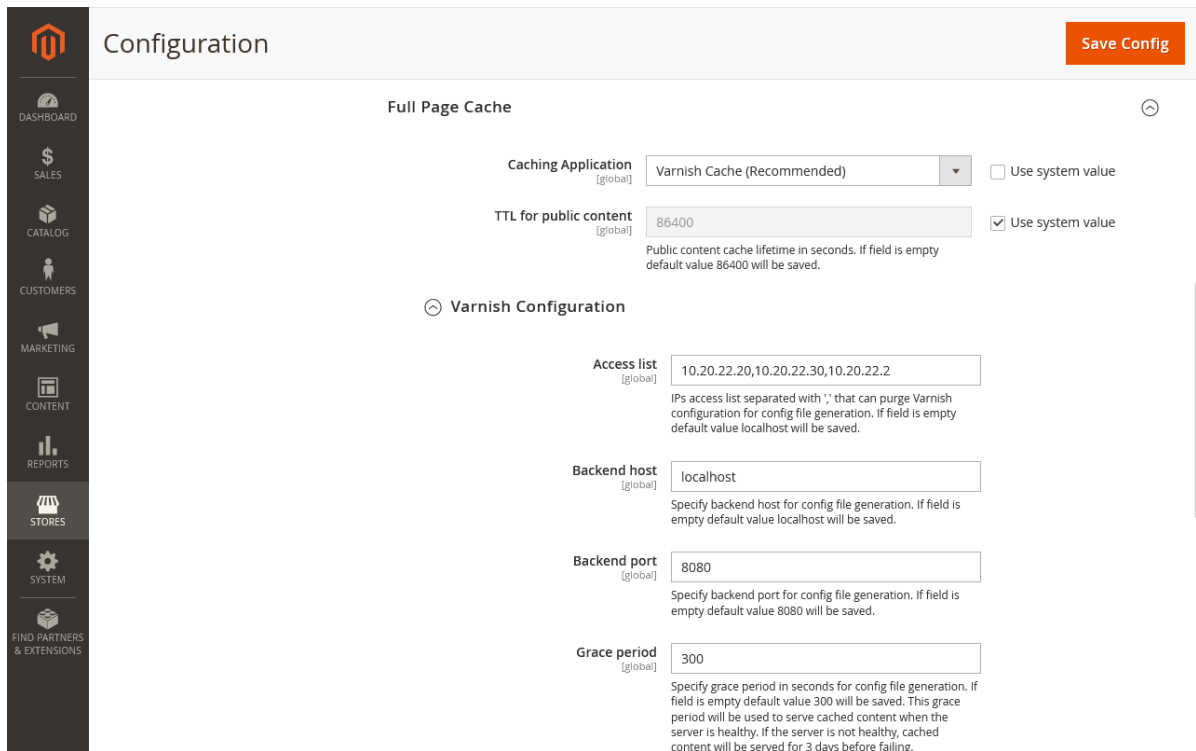


Ilustración 71 - Adobe Commerce Magento *Open source*. Configuración full page caché 1.

Ahora es necesario descargar el archivo de configuración de Varnish compatible con la versión instalada 6.0:

The screenshot shows the 'Configuration' page in Adobe Commerce. The 'Backend port' is set to 8080, and the 'Grace period' is set to 300. There are three buttons for exporting VCL files for Varnish 4, 5, and 6. The sidebar on the left contains navigation icons for Dashboard, Sales, Catalog, Customers, Marketing, Content, Reports, Stores, System, and Find Partners & Extensions. A 'Save Config' button is in the top right corner.

Ilustración 72 - Adobe Commerce Magento Open source. Configuración de full page caché 2.

Descargado este archivo, se deberá subir a los servidores Balanceador, y sustituir con los mismos los archivos “default.vcl” dentro del directorio de configuración de Varnish (/etc/varnish), y a continuación reiniciar el servicio.

Por último, para que el borrado de caché desde consola o desde backoffice sea efectivo, se deberá modificar a mano el archivo de entorno de Magento (/media/E-Commerce/current/app/etc/env.php) para añadir el servidor de Varnish, de forma que las peticiones de borrado de caché lleguen correctamente:

```
'http_cache_hosts' => [
  [
    'host' => '10.20.22.10',
    'port' => '6081'
  ]
]
```

Ilustración 73 - Adobe Commerce Magento Open source. Configuración de full page caché 3.

Es necesario también indicar a magento que debe usar Elasticsearch como motor de búsqueda, para lo que se debe acceder a la configuración en el *backend* a Stores -> Configuration -> Catalog, en la sección Catalog Search e indicarle al aplicativo los valores, aunque ya fueran indicados en la cadena de instalación, y realizar un test de conexión:

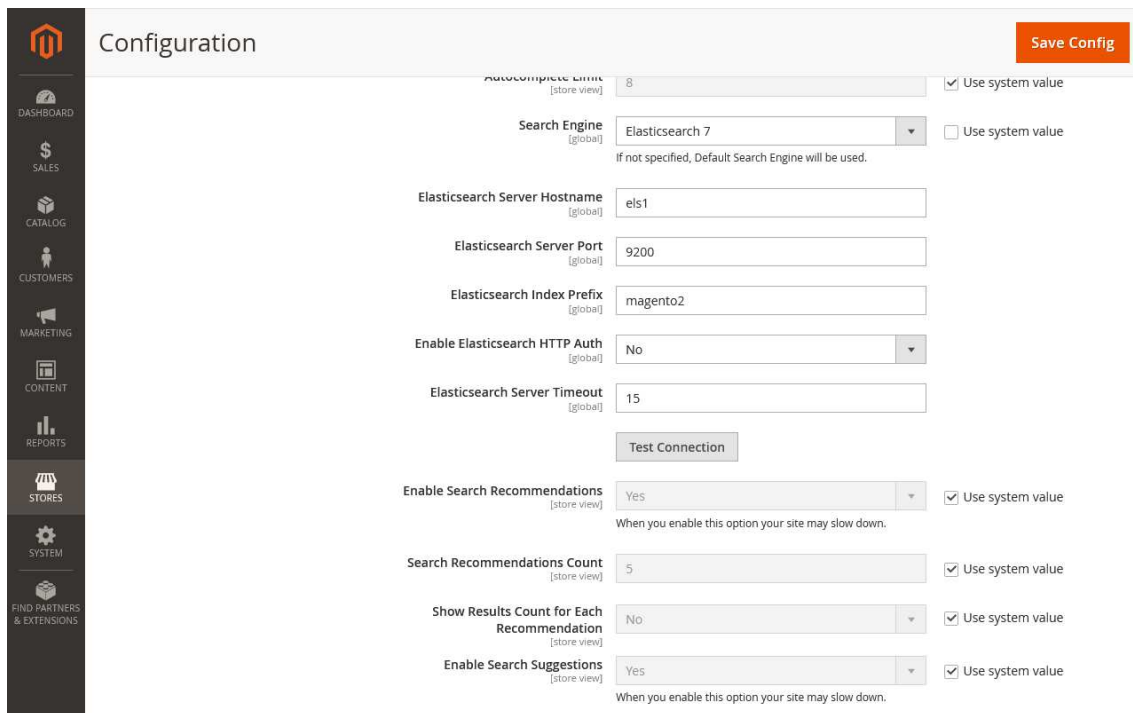


Ilustración 74 - Adobe Commerce Magento *Open source*. Configuración Store Search Engine.

En este punto de inflexión tenemos el *E-Commerce* Adobe Commerce Magento *Open source*, instalado y distribuido:

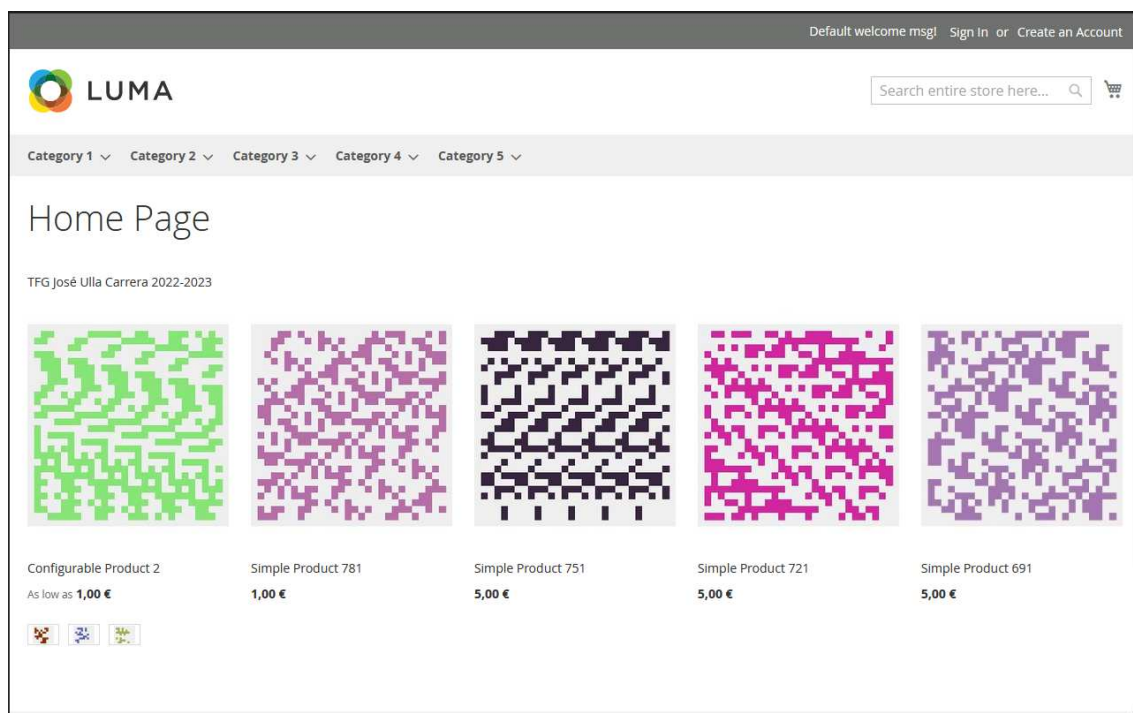


Ilustración 75 - Adobe Commerce Magento *Open source*. Frontend.

5. Alta disponibilidad y tolerancia a fallos

Con la arquitectura y el software *ecommerce* instalado, la siguiente etapa consiste en dotar al aplicativo *ecommerce* de alta disponibilidad y

tolerancia a fallos, debido a las limitaciones de tiempo y hardware, se tratará de conseguir un entorno tolerante a fallos, que no lograría esta alta disponibilidad de forma automática, pero conseguirá no obstante tener tiempos de parada o indisponibilidad mínimos.

Lograr alta disponibilidad en un entorno distribuido puede parecer simple de forma teórica, al fin y al cabo, consiste replicar las máquinas de cada uno de los servicios y configurar la arquitectura en consecuencia para usar todas las instancias, o para que unas instancias estén activas y las otras en espera ante un evento disruptivo en las primeras.

Esto presenta una dicotomía importante que valorar y que se debe decidir a nivel de servicio, pero que a su vez permita funcionar el global de la arquitectura, se presenta entonces la decisión: replicación o clusterización.

Mediante replicación se obtiene alta disponibilidad y tolerancia a fallos, sin embargo, mediante clusterización es posible obtener ambas, aunque en muchos casos únicamente una u otra. Además, hay servicios que no se pueden replicar, tales como los transaccionales o generalizando cualquier SGBD, y otros cuya clusterización requeriría del pago de licencias de software adicionales, algo que se trata de evitar en este trabajo.

La replicación es simple, se aumenta el número de instancias del servicio en cuestión, y se modifica la configuración para que se usen todas las réplicas. Esto supone dos problemas fundamentalmente, el control de las réplicas y la dificultad de automatizar o prevenir los eventos disruptivos, por lo que en el marco de este trabajo estas casuísticas y las tareas pertinentes quedan relegadas a un enfoque totalmente reactivo: las instancias de los servicios y la configuración de la arquitectura se hacen de forma manual por un administrador. Se usará la replicación de servicios para el routing, el balanceo, el caché de página y los intérpretes de PHP.

Por el contrario, la opción de clusterización contiene mayor dificultad ya que depende fuertemente del servicio a obtener alta disponibilidad y/o tolerancia a fallos. Hay servicios cuya clusterización es relativamente básica y sólo confieren un mecanismo de replicación activo, de forma que, ante un evento disruptivo del servidor principal, la suplantación del mismo por la réplica requiere de una serie de tareas adicionales, en este caso concreto nos encontramos con el servicio de Base de Datos relacional con MariaDB, es decir obtendremos tolerancia a fallos, pero no alta disponibilidad. Por otro lado, hay servicios que sí permiten obtener ambas mediante el uso de clústeres activos, en este trabajo tanto el servicio de caché Redis como el de búsquedas Elasticsearch pertenecen a esta clasificación.

5.1. Replicación de rutas

Hasta este capítulo existía una máquina de infraestructura router, con una conexión WAN y otra LAN, la primera provee tanto de acceso a internet como punto de entrada utilizando NAT y redireccionamiento de puertos. Este router pasará a tener una segunda conexión WAN2 que proporcionará una

segunda IP que se usará para proporcionar acceso al *E-Commerce* a través de un segundo balanceador y un segundo servicio de *full page cache*.

El motivo de esta configuración es debido a las limitaciones de hardware, teniendo una única interfaz de red física, y estando ya en un entorno virtualizado con un único hipervisor no se justifica una segunda máquina router completa. Si bien, en un entorno de producción real, las líneas han de ser físicamente redundantes y entrar en el centro de datos por rutas diferentes, deben proporcionarse dos routers diferentes, conectados a fases eléctricas diferentes y conectarse a una red con mecanismos de redundancia etc.

Si el e-commerce se encuentra en una nube pública también ha de tener una replicación de rutas, pero duplicando las máquinas virtuales encargadas del acceso externo, o bien usar un servicio IaaS público, como por ejemplo el que ofrece Amazon *Web Services* con AWS EC2 Load Balancer.

Retomando el marco de trabajo, la máquina router es configurada con una nueva conexión de red virtual a la red física:

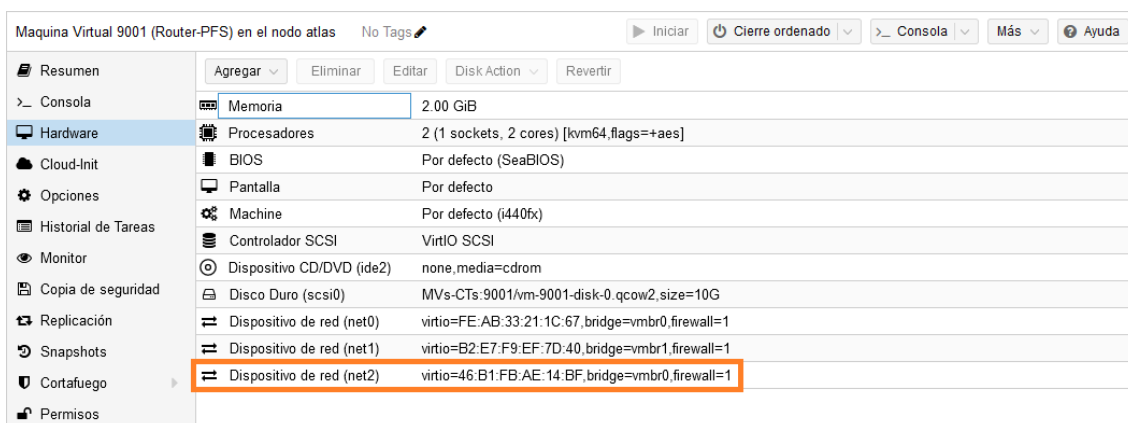


Ilustración 76 - Configuración de red de VM Router.

Esta segunda conexión de red obtiene una IP de red local, que en este caso emula una conexión a internet, la cual se añade al registro DNS local de la red para proporcionar la redundancia al nombre DNS del *E-Commerce*:

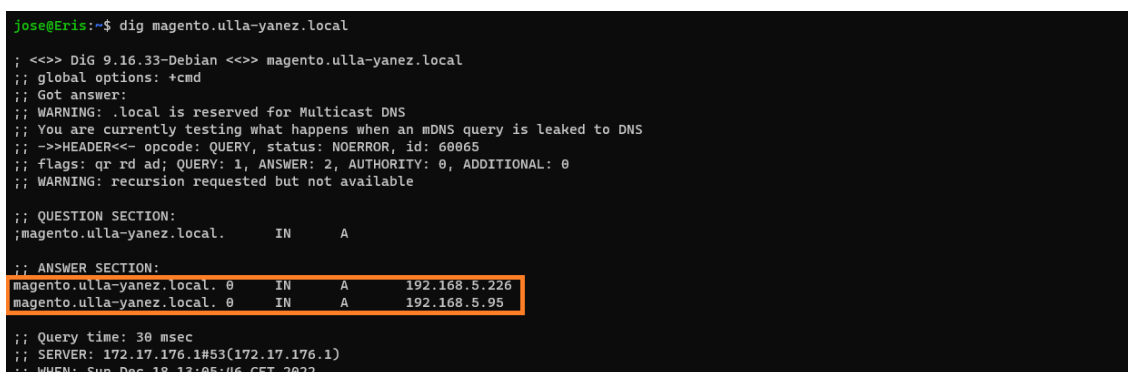


Ilustración 77 - Redundancia de rutas.

Por lo que el *E-Commerce* estará accesible desde cualquiera de esas IP, proporcionando alta disponibilidad:

Firewall / NAT / Port Forward

Port Forward 1:1 Outbound NPT

Rules	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions
<input type="checkbox"/>	<input checked="" type="checkbox"/> WAN2	TCP	*	*	WAN2 address	443 (HTTPS)	10.20.22.11	443 (HTTPS)	NAT Redireccion HTTPS a Nginx LB2	
<input type="checkbox"/>	<input checked="" type="checkbox"/> WAN2	TCP	*	*	WAN2 address	80 (HTTP)	10.20.22.11	80 (HTTP)	NAT Redireccion HTTP a Nginx LB2	
<input type="checkbox"/>	<input checked="" type="checkbox"/> WAN	TCP	*	*	WAN address	443 (HTTPS)	10.20.22.10	443 (HTTPS)	NAT Redireccion HTTPS a Nginx LB1	
<input type="checkbox"/>	<input checked="" type="checkbox"/> WAN	TCP	*	*	WAN address	80 (HTTP)	10.20.22.10	80 (HTTP)	NAT Redireccion HTTP a Nginx LB1	
<input type="checkbox"/>	<input checked="" type="checkbox"/> WAN	TCP	*	*	WAN address	5900 (VNC)	10.20.22.2	5900 (VNC)	VNC a Control	
<input type="checkbox"/>	<input checked="" type="checkbox"/> WAN	TCP	*	*	WAN address	22 (SSH)	10.20.22.2	22 (SSH)	SSH a Control	

Ilustración 78 - Configuración NAT routing final.

5.2. Redundancia de balanceo y caché de página

Este requisito es fácilmente el más simple de proporcionar en lo que a configuración se refiere, teniendo de antemano una segunda IP externa y un router que tenga configurado el NAT y redireccionamiento de puertos correspondiente, como se puede observar en la Ilustración 76 del apartado anterior, únicamente hay que duplicar la máquina balanceadora original, y configurarle su IP interna correspondiente, la configuración de los servicios de balanceo Nginx, o caché de página Varnish no varía.

Conjunto de Recursos: Webservers

Resumen

Miembros	Tipo ↑	Descripción	Uso de dis...	Memoria - ...	Uso de CPU	Tiempo de uso	Host CPU ...	Host Mem...
<input checked="" type="checkbox"/>	lxc	9010 (LB1-D11)	12.5 %	33.7 %	0.2% of 2 ...	2 días 12:32:13	0.1% of 8C...	1.1 %
<input checked="" type="checkbox"/>	lxc	9011 (LB2-D11)	12.4 %	6.6 %	0.2% of 2 ...	5 días 20:56:01	0.1% of 8C...	0.2 %

Ilustración 79 - Configuración balanceadores final.

Sin embargo, es necesario indicarle a Magento que se ha modificado la cantidad de servidores que realizan el cacheo de página, por lo que al igual que en la sección anterior es necesario modificar los parámetros en la configuración en Stores -> Configuration -> Advanced -> System, en la sección Full Page Cache, modificar las IP de los hosts que pueden acceder al purgado, añadiendo los dos nuevos intérpretes de PHP de la sección siguiente:

The screenshot shows the 'Configuration' page for Varnish. On the left is a sidebar with navigation icons for Dashboard, Sales, Catalog, Customers, Marketing, Content, Reports, Stores, System, and Find Partners & Extensions. The main content area is titled 'Configuration' and includes a 'Save Config' button. Below this, there are several configuration fields:

- TTL for public content** [global]: 86400. A checkbox 'Use system value' is checked. Below the field, it says: 'Public content cache lifetime in seconds. If field is empty default value 86400 will be saved.'
- Varnish Configuration** (collapse icon):
 - Access list** [global]: 10.20.22.20,10.20.22.21,10.20.22.30,10.20.22.31. Below: 'IPs access list separated with ',' that can purge Varnish configuration for config file generation. If field is empty default value localhost will be saved.'
 - Backend host** [global]: localhost. Below: 'Specify backend host for config file generation. If field is empty default value localhost will be saved.'
 - Backend port** [global]: 8080. Below: 'Specify backend port for config file generation. If field is empty default value 8080 will be saved.'
 - Grace period** [global]: 300. Below: 'Specify grace period in seconds for config file generation. If field is empty default value 300 will be saved. This grace period will be used to serve cached content when the server is healthy. If the server is not healthy, cached content will be served for 3 days before falling.'
- Export Configuration** [global]: Two buttons: 'Export VCL for Varnish 4' and 'Export VCL for Varnish 5'.

Ilustración 80 - Configuración de caché de página final 1.

Posteriormente se debe exportar el VCL, copiarlo en los balanceadores y reiniciar el servicio, al igual que se realizó en la instalación inicial distribuida.

Y de modo manual, en el archivo de configuración del entorno, es necesario indicarle que hay dos host que sirven caché de página completa, por lo que se debe añadir al array correspondiente las IP de ambos balanceadores:

```
'http_cache_hosts' => [
  [
    'host' => '10.20.22.10',
    'port' => '6881',
    'host' => '10.20.22.11',
    'port' => '6881'
  ]
]
```

Ilustración 81 - Configuración de caché de página final 2.

5.3. Redundancia de intérpretes de PHP

Para poder llegar a tener redundancia en los intérpretes de PHP es necesario, por un lado, replicar tantas máquinas como intérpretes sean necesarios en el pool, configurar sus correspondientes IPs en el sistema operativo o hipervisor y, además, modificar la configuración en los balanceadores para que se envíen peticiones a estos.

Debido a las restricciones de recursos físicos, se duplican ambas máquinas intérpretes de PHP, de forma que tendremos una configuración de 2 + 2 intérpretes, para *frontend* y *backoffice* respectivamente:

Conjunto de Recursos: WorkersPHP								
Resumen		Agregar	Eliminar					
Miembros	Tipo ↑	Descripción	Uso de dis...	Memoria - ...	Uso de CPU	Tiempo de uso	Host CPU ...	Host Mem...
lxc	lxc	9020 (FE1-D11)	5.6 %	3.4 %	0.0% of 2 ...	2 días 12:36:12	0.0% of 8C...	0.4 %
lxc	lxc	9021 (FE1-D11)	5.6 %	2.7 %	0.0% of 2 ...	01:30:48	0.0% of 8C...	0.3 %
lxc	lxc	9030 (BE1-D11)	5.7 %	4.1 %	0.0% of 2 ...	2 días 12:36:04	0.0% of 8C...	0.5 %
lxc	lxc	9031 (BE1-D11)	5.6 %	3.5 %	0.0% of 2 ...	01:30:29	0.0% of 8C...	0.4 %

Ilustración 82 - Configuración intérpretes PHP final.

A continuación, se modifica en el archivo de configuración del balanceador interno, la configuración de cada uno de los *upstreams* correspondientes a *frontend* y *backoffice*, esto se hace en ambos balanceadores en la infraestructura final:

```
root@LB2-D11:~# head /etc/nginx/sites-enabled/int-magento.conf
# Workers para front
upstream frontend {
    server 10.20.22.20:9000;
    server 10.20.22.21:9000;
}
# Workers para back
upstream backend {
    server 10.20.22.30:9000;
    server 10.20.22.31:9000;
}
```

Ilustración 83 - Configuración balanceo final.

La configuración del servicio de *full page cache* no sufre ninguna modificación.

En este punto están configurados todos los servicios que obtienen alta disponibilidad y tolerancia a fallos mediante replicación, a continuación, se realiza la configuración de los servicios que las obtendrán mediante *clustering*.

5.4. Clúster de base de datos relacional MariaDB

Esta configuración constará de dos fases, en una primera fase se clonará el servidor actual de SGBD en una copia exacta, configurará su IP correspondiente para posteriormente modificar la configuración de ambos servidores para que el servidor primario acepte las conexiones de replicación por parte del servidor secundario.

En el archivo de configuración de MariaDB en `/etc/mysql/my.cnf` se debe buscar la sección de replicación, modificar y descomentar las siguientes líneas, es especialmente importante modificar el `server-id` y el `report-host` para que la replicación funcione sin problemas:

```
#log-queries-not-using-indexes
#log_slow_admin_statements
#
# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replication slave, see README.Debian about
# other settings you may need to change.
server-id = 1
report_host = SGBD1
#auto_increment_increment = 2
#auto_increment_offset = 1
log_bin = /var/log/mysql/mariadb-bin
log_bin_index = /var/log/mysql/mariadb-bin.index
# not fab for performance, but safer
#sync_binlog = 1
expire_logs_days = 10
max_binlog_size = 100M
# slave
relay_log = /var/log/mysql/relay-bin
relay_log_index = /var/log/mysql/relay-bin.index
#relay_log_info_file = /var/log/mysql/relay-bin.info
#log_slave_updates
#read_only
```

Ilustración 84 - Replicación MariaDB - Configuración servidor activo.

Posteriormente se reinicia el servicio de MariaDB y se continúa con la configuración en el propio SGBD, desde el cliente MariaDB se añade un usuario el cual podrá ejecutar la replicación, se configuran sus permisos y recarga la tabla de privilegios:

```
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 36
Server version: 10.4.27-MariaDB-1:10.4.27+maria-deb10-log mariadb.org binary distribution

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 USER 'replica'@'%' IDENTIFIED BY 'izEBMLGLi023knM';
Query OK, 0 rows affected (0.005 sec)

MariaDB [(none)]> GRANT REPLICATION SLAVE ON *.* TO 'replica'@'%';
Query OK, 0 rows affected (0.005 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> |
```

Ilustración 85 - Replicación MariaDB - Configuración permisos.

Podemos anotar la posición en la que se encuentra la replicación:

```
MariaDB [(none)]> SHOW MASTER STATUS;
+-----+-----+-----+-----+
| File | Position | BinLog_Do_DB | BinLog_Ignore_DB |
+-----+-----+-----+-----+
| mariadb-bin.000004 | 795 | | |
+-----+-----+-----+-----+
1 row in set (0.000 sec)
```

Ilustración 86 - Replicación MariaDB - Estado inicial servidor activo.

Es importante destacar que todo este proceso ha de hacerse con la escritura en la base de datos deshabilitada, dado que un cambio en el log de replicación podrá hacer que el proceso de configuración falle.

Además hay un paso extra necesario en caso de empezar con un servidor pasivo SGBD recién instalado, y es hacer una descarga de todo el contenido de la base de datos durante este tiempo de deshabilitado de escritura, para poder cargar en el servidor pasivo. En este ejemplo no es necesario porque se parte de un clon del servidor activo, ya con la escritura deshabilitada, por lo que la base de datos ya se encuentra precargada y en estado consistente.

Se continúa la configuración en el servidor pasivo, o servidores en caso de configurar varios, modificando al igual que en el activo en el archivo de configuración en /etc/mysql/my.cnf se debe buscar la sección de replicación,

modificar y descomentar las siguientes líneas, es especialmente importante modificar el server-id y el report-host para que la replicación funcione sin problemas

```
#
# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replication slave, see README.Debian about
# other settings you may need to change.
server-id             = 2
report_host          = SGBD2
#auto_increment_increment = 2
#auto_increment_offset  = 1
log_bin              = /var/log/mysql/mariadb-bin
log_bin_index        = /var/log/mysql/mariadb-bin.index
# not fab for performance, but safer
#sync_binlog         = 1
expire_logs_days     = 10
max_binlog_size      = 100M
# slaves
relay_log            = /var/log/mysql/relay-bin
relay_log_index      = /var/log/mysql/relay-bin.index
#relay_log_info_file  = /var/log/mysql/relay-bin.info
#log_slave_updates
#read_only
```

Ilustración 87 - Replicación MariaDB - Configuración servidor pasivo.

Se para la replicación, este paso es esencial para la replicación, es un punto de fallo común a la hora de realizar este tipo de configuraciones:

```
MariaDB [(none)]> STOP SLAVE;
Query OK, 0 rows affected, 1 warning (0.000 sec)
```

Ilustración 88 - Replicación MariaDB - Parada de la replicación.

Se configura el origen de replicación, el servidor activo y el usuario creado previamente que tiene los privilegios para realizarla y se inicia la replicación:

```
MariaDB [(none)]> CHANGE MASTER TO MASTER_HOST='10.20.22.40', MASTER_USER='replica', MASTER_PASSWORD = 'izeBMLGLi023knM', MASTER_LOG_FILE='mariadb-bin.000004', MASTER_LOG_POS=795;
Query OK, 0 rows affected (0.064 sec)

MariaDB [(none)]> START SLAVE;
Query OK, 0 rows affected (0.000 sec)
```

Ilustración 89 - Replicación MariaDB - Configuración y arranque de replicación.

En la siguiente figura podemos observar el estado de replicación del servidor pasivo una vez reiniciado el proceso de replicación:


```

MariaDB [(none)]> SHOW SLAVE STATUS\G;
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: 10.20.22.40
Master_User: replica
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mariadb-bin.000004
Read_Master_Log_Pos: 795
Relay_Log_File: relay-bin.000002
Relay_Log_Pos: 557
Relay_Master_Log_File: mariadb-bin.000004
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB:
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 795
Relay_Log_Space: 860
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 1
Master_SSL_Crl:
Master_SSL_Crlpath:
Using_Gtid: No
Gtid_IO_Pos:
Replicate_Do_Domain_Ids:
Replicate_Ignore_Domain_Ids:
Parallel_Mode: conservative
SQL_Delay: 0
SQL_Remaining_Delay: NULL
Slave_SQL_Running_State: Slave has read all relay log; waiting for the slave I/O thread to update it
Slave_DDL_Groups: 0
Slave_Non_Transactional_Groups: 0
Slave_Transactional_Groups: 0
1 row in set (0.000 sec)

```

Ilustración 90 - Replicación MariaDB - Estado inicial.

En esta figura podemos observar la posición del log binario antes y después de la modificación del precio de un producto:

```

MariaDB [(none)]> SHOW MASTER STATUS;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mariadb-bin.000004 | 795     |              |                  |
+-----+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [(none)]> SHOW MASTER STATUS;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mariadb-bin.000004 | 65387   |              |                  |
+-----+-----+-----+-----+
1 row in set (0.000 sec)

```

Ilustración 91 - Replicación MariaDB - Estado de logs tras un cambio en Backoffice.

Y como de forma automática el servidor pasivo obtiene los cambios mediante la replicación:

```
MariaDB [(none)]> SHOW SLAVE STATUS\G;
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: 10.20.22.40
Master_User: replica
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mariadb-bin.000004
Read_Master_Log_Pos: 65387
Relay_Log_File: relay-bin.000002
Relay_Log_Pos: 52823
Relay_Master_Log_File: mariadb-bin.000004
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB:
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 65387
Relay_Log_Space: 53126
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 1
Master_SSL_Crl:
Master_SSL_Crlpath:
Using_Gtid: No
Gtid_IO_Pos:
Replicate_Do_Domain_Ids:
Replicate_Ignore_Domain_Ids:
Parallel_Mode: conservative
SQL_Delay: 0
SQL_Remaining_Delay: NULL
Slave_SQL_Running_State: Slave has read all relay log; waiting for the slave I/O thread to update it
Slave_DDL_Groups: 3
Slave_Non_Transactional_Groups: 4
Slave_Transactional_Groups: 14
1 row in set (0.000 sec)
```

Ilustración 92 - Replicación MariaDB - Replicación exitosa.

Esta configuración no confiere alta disponibilidad en el servicio de base de datos, pero si confiere tolerancia a fallos, de forma que ante un evento disruptivo en el SGBD activo, el servidor pasivo siempre tendrá una copia exacta de los datos transaccionalmente consistente, de modo que se podrá promocionar a servidor principal, o se podrá usar de base de datos de sólo lectura mientras se realiza la recuperación del servidor activo.

5.5. Clúster de base de datos de objetos Redis

Esta configuración es muy similar a la del anterior SGBD, debido a las limitaciones de hardware, concretamente de memoria y núcleos de procesamiento, se propone una configuración básica de servicio Redis en modo activo-pasivo, comenzando por la restauración de una copia exacta del servidor actual, a la cual se le configurará su IP correspondiente.

A continuación, es necesario hacer un único cambio en la configuración del servicio de Redis, del servidor secundario, añadiendo a su archivo de configuración en /etc/redis/redis.conf:

```

root@RDS2-D11:~# tail /etc/redis/redis.conf
# Set bgsave child process to cpu affinity 1,10,11
# bgsave_cpulist 1,10-11

# In some cases redis will emit warnings and even refuse to start if it detects
# that the system is in bad state, it is possible to suppress these warnings
# by setting the following config which takes a space delimited list of warnings
# to suppress
#
# ignore-warnings ARM64-COW-BUG
slaveof 10.20.22.50 6379

```

Ilustración 93 - Replicación Redis - Configuración servidor secundario.

Reiniciar el servicio y comprobar que la replicación está activa en el servidor primario:

```

root@RDS1-D11:~# redis-cli
127.0.0.1:6379> info replication
# Replication
role:master
connected_slaves:1
slave0:ip=10.20.22.51,port=6379,state=online,offset=762935,lag=0
master_failover_state:no-failover
master_replid:c43ffdf18ac700988108f26bf67e9a05848a6087
master_replid2:0000000000000000000000000000000000000000
master_repl_offset:762935
second_repl_offset:-1
repl_backlog_active:1
repl_backlog_size:1048576
repl_backlog_first_byte_offset:1
repl_backlog_histlen:762935
127.0.0.1:6379> |

```

Ilustración 94 - Replicación Redis - Estado primario.

Y en el secundario:

```

root@RDS2-D11:~# redis-cli
127.0.0.1:6379> info replication
# Replication
role:slave
master_host:10.20.22.50
master_port:6379
master_link_status:up
master_last_io_seconds_ago:5
master_sync_in_progress:0
slave_read_repl_offset:762921
slave_repl_offset:762921
slave_priority:100
slave_read_only:1
replica_announced:1
connected_slaves:0
master_failover_state:no-failover
master_replid:c43ffdf18ac700988108f26bf67e9a05848a6087
master_replid2:0000000000000000000000000000000000000000
master_repl_offset:762921
second_repl_offset:-1
repl_backlog_active:1
repl_backlog_size:1048576
repl_backlog_first_byte_offset:1
repl_backlog_histlen:762921
127.0.0.1:6379> |

```

Ilustración 95 - Replicación Redis - Estado secundario.

Esta configuración, al igual que en SGBD MariaDB no confiere alta disponibilidad, pero si tolerancia a fallos. También es posible promocionar el servidor secundario fácilmente, por lo que un administrador podría cambiar en caliente la configuración sin pérdida de servicio, o podría cambiar la configuración en caso de interrupción del servidor primario en muy poco tiempo.

5.6. Clúster de servicio de búsqueda Elasticsearch

Elasticsearch no permite un clúster básico activo-pasivo, dado a que por defecto debe de poder evitar las situaciones de Split-brain, es decir que debe haber un mínimo de tres nodos, de los cuales dos deben ser elegibles como principales, de forma que siempre haya quorum de dónde se encuentre la información más reciente.

Esta situación complica un poco la configuración, se partirá de tres nodos, dos de ellos réplicas exactas del nodo único inicial, por lo que el aplicativo Elasticsearch ya se encontrará instalado y únicamente será necesario configurar las IP correspondientes a cada nodo y modificar las configuraciones de cada uno de los mismos para formar el clúster.

Para ello se modifica en la configuración de Elasticsearch en /etc/Elasticsearch/Elasticsearch.xml, para cada uno de los tres contenedores miembros del clúster:

Nodo els1:

```
root@ELS1-D11:~# cat /etc/elasticsearch/elasticsearch.yml | grep -v '#'  
cluster.name: els-magento  
node.master: true  
node.data: true  
node.name: els1  
path.data: /var/lib/elasticsearch  
path.logs: /var/log/elasticsearch  
network.host: 0.0.0.0  
http.port: 9200  
discovery.seed_hosts: ["els1", "els2", "els3"]  
cluster.initial_master_nodes: ["els1", "els2"]
```

Ilustración 96 - Clúster Elasticsearch - Configuración nodo els1.

Nodo els2:

```
root@ELS2-D11:~# cat /etc/elasticsearch/elasticsearch.yml | grep -v '#'  
cluster.name: els-magento  
node.master: true  
node.data: true  
node.name: els2  
path.data: /var/lib/elasticsearch  
path.logs: /var/log/elasticsearch  
network.host: 0.0.0.0  
http.port: 9200  
discovery.seed_hosts: ["els1", "els2", "els3"]  
cluster.initial_master_nodes: ["els1", "els2"]
```

Ilustración 97 - Clúster Elasticsearch - Configuración nodo els2.

Nodo els3:

```
root@ELS3-D11:~# cat /etc/elasticsearch/elasticsearch.yml | grep -v '#'  
cluster.name: els-magento  
node.master: true  
node.data: true  
node.name: els3  
path.data: /var/lib/elasticsearch  
path.logs: /var/log/elasticsearch  
network.host: 0.0.0.0  
http.port: 9200  
discovery.seed_hosts: ["els1", "els2", "els3"]  
cluster.initial_master_nodes: ["els1", "els2"]
```

Ilustración 98 - Clúster Elasticsearch - Configuración nodo els3.

Debido a la formación de un clúster, el software endurece las condiciones de configuración de la JVM, por lo que con la configuración de un único host de la que derivan los tres nodos, al reiniciar los servicios darán error, esto se corrige modificando los parámetros de JVM para hacer coincidir los valores -Xms y -Xmx as 3 Gigabytes, por lo que para cada uno de los nodos se modificará la configuración de la JVM en /etc/Elasticsearch/jvm.options:

```

#####
## IMPORTANT: JVM heap size
#####
##
## The heap size is automatically configured by Elasticsearch
## based on the available memory in your system and the roles
## each node is configured to fulfill. If specifying heap is
## required, it should be done through a file in jvm.options.d,
## and the min and max should be set to the same value. For
## example, to set the heap to 4 GB, create a new file in the
## jvm.options.d directory containing these lines:
##
##
-Xms3g
-Xmx3g

```

Ilustración 99 - Clúster Elasticsearch - Modificación parámetros JVM.

Finalmente, todos los contenedores son clones del inicial, por lo que a efectos de cluster, todos tiene el mismo identificador, por lo que aunque el cluster parezca arrancar correctamente, realmente cada uno de los nodos estaría formando un cluster de un único nodo y no habría comunicación entre ellos, por lo que es necesario eliminar la carpeta /var/lib/Elasticsearch/nodes, en al menos dos de ellos, de esta forma se generan identificadores nuevos al reiniciar el servicio.

Tras reiniciar los servicios en los tres nodos del cluster de Elasticsearch, es posible comprobar que el cluster se ha formado correctamente haciendo una llamada al puerto en el que el servicio atiende peticiones:

```

root@control:~# curl -XGET 'http://10.20.22.60:9200/_cluster/health'
{"cluster_name":"els-magento","status":"green","timed_out":false,"number_of_nodes":3,"number_of_data_nodes":3,"active_primary_shards":4,"active_shards":8,"relocating_shards":0,"initializing_shards":0,"unassigned_shards":0,"delayed_unassigned_shards":0,"number_of_pending_tasks":0,"number_of_in_flight_fetch":0,"task_max_waiting_in_queue_millis":0,"active_shards_percent_as_number":100.0}root@control:~#

```

Ilustración 100 - Clúster Elasticsearch - Comprobación del clúster.

Dónde se observa que el cluster está formado con 3 nodos y en estado *green*, es decir funcionando de forma óptima.

Esta configuración otorga alta disponibilidad y tolerancia a fallos, sin embargo Magento no contempla la configuración de varios servidores de Elasticsearch, y asimismo Elasticsearch no proporciona un mecanismo de IP flotante o similar que permita comunicar a nivel de clúster, por lo que se adopta una solución idéntica al clúster de almacenamiento, aunque no sea óptima, de crear un nombre DNS al cluster que resuelva las tres direcciones IP del cluster, y configurar éste en el backoffice de Magento:

Host Overrides				
Host	Parent domain of host	IP to return for host	Description	Actions
els	tfg.local	10.20.22.60,10.20.22.61,10.20.22.62	Resolucion cluster elasticsearch	 
str	tfg.local	10.20.22.70,10.20.22.71,10.20.22.72	Resolucion cluster almacenamiento	 

Ilustración 101 - Configuración DNS clúster Elasticsearch.

Con lo que se configurará en el backoffice de Magento en Stores -> Configuration -> Catalog, en la sección Catalog Search:

The screenshot shows the 'Configuration' page for Elasticsearch in Magento. The left sidebar contains navigation icons for Dashboard, Sales, Catalog, Customers, Marketing, Content, Reports, Stores, System, and Find Partners & Extensions. The main content area displays the following configuration options:

- Autocomplete Limit** [store view]: 8 Use system value
- Search Engine** [global]: Elasticsearch 7 Use system value
If not specified, Default Search Engine will be used.
- Elasticsearch Server Hostname** [global]: els
- Elasticsearch Server Port** [global]: 9200
- Elasticsearch Index Prefix** [global]: magento2
- Enable Elasticsearch HTTP Auth** [global]: No
- Elasticsearch Server Timeout** [global]: 15
- Successful! Test again?** (button)
- Enable Search Recommendations** [store view]: Yes Use system value
When you enable this option your site may slow down.
- Search Recommendations Count** [store view]: 5 Use system value
- Show Results Count for Each Recommendation** [store view]: No Use system value
- Enable Search Suggestions** [store view]: Yes Use system value
When you enable this option your site may slow down.

Ilustración 102 - Configuración final Elasticsearch Magento.

5.7. Ilustración final de la infraestructura.

En la ilustración final de la infraestructura se puede observar la infraestructura desarrollada en este trabajo, con sus partes diferenciadas distribuidas, así como la replicación de estas para conferir la alta disponibilidad y/o tolerancia a fallos.

En la página siguientes se encuentra una ilustración a tamaño completo con un gráfico del estado final de la infraestructura.

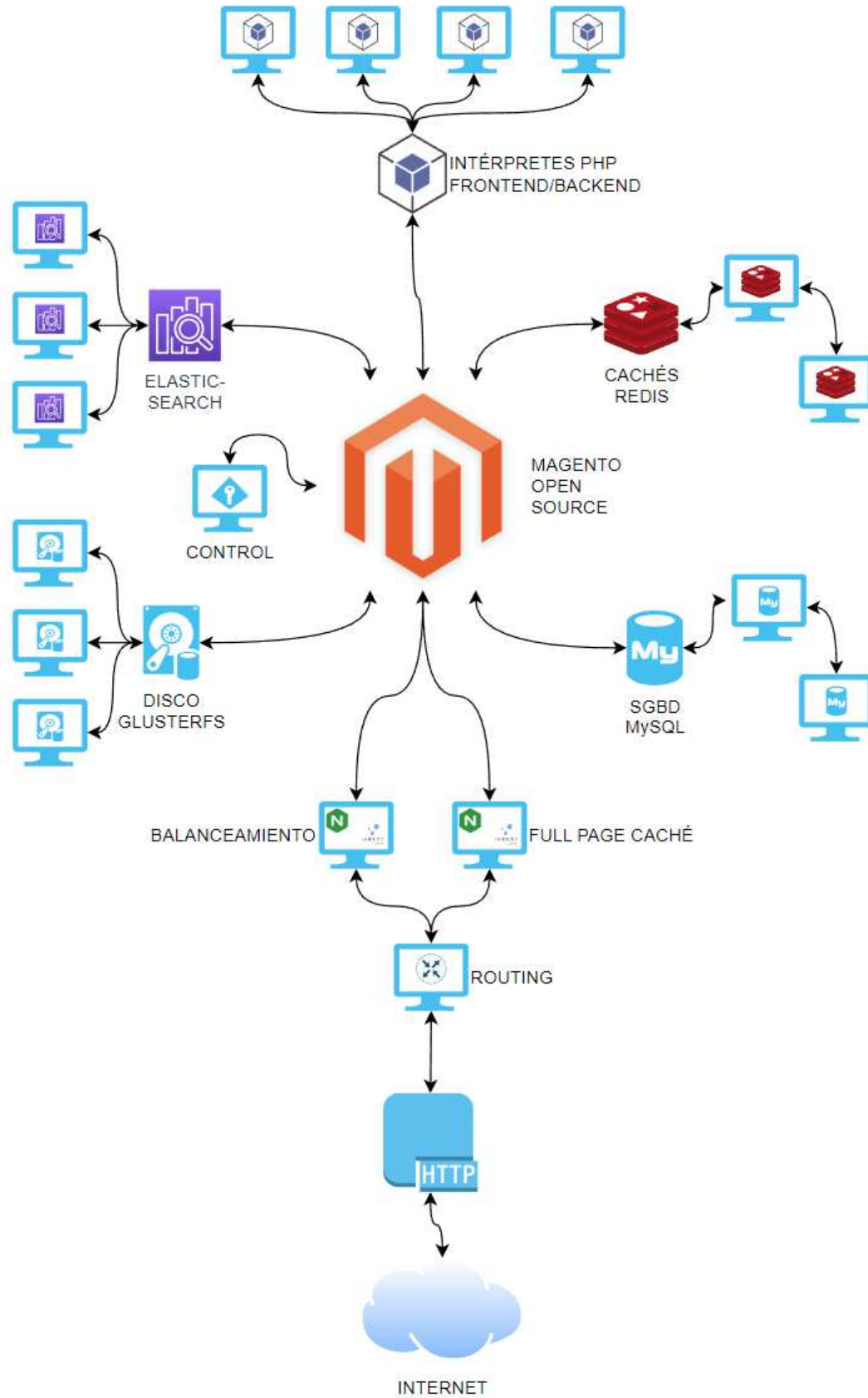


Ilustración 103 - Arquitectura final.

6. Conclusiones

Este trabajo ha servido para definir una arquitectura distribuida, escalable y tolerante a fallos, adaptable a prácticamente cualquier aplicación que utilice un *stack* LEMP, con mínimas variaciones, de forma que es válida para un E-Commerce como Magento, pero también para un E-Commerce Prestashop, así como para un CMS como Wordpress o Joomla, LMS como Moodle Y aplicaciones colaborativas como Nextcloud u Owncloud.

También ha servido para construir una plataforma de virtualización con capacidad para ejecutar máquinas virtuales y contenedores, con un sistema simple de gestión vía web usando hardware de propósito general, conteniendo al extremo la inversión en equipamiento.

Otros aspectos importantes, aunque no se ha profundizado en ellos, son la administración de una red virtual y el uso de *appliance* virtuales tipo PFSense para la gestión de los servicios de red, reduciendo la complejidad de la red sobre la que se sustenta la arquitectura.

También se han realizado configuraciones necesarias para salvaguardar la información de la aplicación, emulando un sistema en producción utilizando un sistema de backups cohesionado con Proxmox el cual se detalla en el anexo 9.1 de este documento.

Un aspecto importante de este trabajo que es necesario destacar es que ha estado siempre guiado desde un punto de vista libre y comunitario, ningún software usado en este trabajo está gravado con licencias, cuotas o gastos derivados de ningún tipo, si bien los mantenedores proveen de servicios de mantenimiento y licenciamiento con funcionalidades premium o soporte técnico dedicado. Esta característica resulta en una arquitectura, tanto virtual, como de aplicación con un coste muy reducido, asequible para economías de cualquier tipo.

Se puede considerar que el proyecto ha sido un éxito y se ha obtenido un producto que cumple las especificaciones iniciales, aunque con amplio margen de mejora en bastantes aspectos, detallados en el punto siguiente de este capítulo.

En lo que a planificación se refiere, los objetivos han sido razonables para la limitación de tiempo impuesta para el trabajo, y se ha conseguido mantener la planificación esperada, la metodología en cascada es la ideal en un proyecto de este tipo, dado que sólo existe un recurso y es necesario entregar por fases para avanzar en las tareas.

Ha habido dos incidentes que han impactado en la planificación, pero que se han podido corregir. Por un lado, la elección inicial del sistema de

almacenamiento (Ceph), el cual tenía ciertos requisitos de infraestructura que no se podrían cumplir con el hardware disponible, por lo que se tuvo que replantear esa parte del proyecto y adaptar a otra tecnología. Por otro lado, y en tono más personal: problemas de salud que llevo arrastrando este año desde una operación que me realizaron en febrero.

6.1. Puntos de Mejora

Durante el desarrollo de este trabajo se han detectado varios puntos de mejora susceptibles de realizarse a continuación, si bien únicamente el siguiente es realmente importante para el desempeño de la arquitectura, el resto son coherentes desde un punto de vista de proyecto de forma que nos encontramos:

- **Rendimiento:** aunque el rendimiento del sistema no se mide debido a las capacidades limitadas del hardware, se ha podido observar que hay penalizaciones significativas, especialmente en el almacenamiento. El sistema basado en GlusterFS es altamente dependiente de la velocidad del disco subyacente, en este caso al estar las tres máquinas configuradas en el mismo disco duro, supone un cuello de botella importante a la hora de acceder al dato. Se podría implementar un cambio en la infraestructura de un sistema distribuido a un clúster activo pasivo de almacenamiento basado en TrueNAS, por mantener el espíritu libre del trabajo, sin embargo, esta vía supondría la necesidad de invertir en equipamiento con más RAM.
- **Automatización y alta disponibilidad:** el resultado de la distribución es correcto, sin embargo, la alta disponibilidad no se concluye correctamente, ocurriendo lo mismo con la tolerancia a fallos, casos que deberán contar con un administrador o un equipo de administración que pueda realizar las configuraciones pertinentes cuando ocurran problemas. Idealmente se tratarían de implantar clústeres con alta disponibilidad en todos los servicios que lo permitan, como SGBDs, no obstante, la complejidad de administración a nivel informático aumentaría de manera significativa, tanto en número de instancias a administrar como a nivel de configuración de estas.
- **Auto escalado y orquestamiento:** tampoco se ha contemplado por limitaciones técnicas y de tiempo, pero es posible adaptar esta arquitectura a un sistema con escalado automático. Un sistema que permitiese detectar cuando un servicio está sobrecargado y actuase en consecuencia sería posible de implantar utilizando aplicativos de monitorización como Zabbix, Nagios etc. En conjunto con sistemas de automatización como Puppet, Ansible, Terraform etc. De forma que al detectar una sobrecarga se puedan desplegar horizontalmente nuevas instancias para cubrir la demanda. Del mismo modo un sistema similar podría detectar bajadas de uso y actuando a la inversa generar ahorros significativos en infraestructura,

ya sea apagando servidores en modo On-Premises o instancias de servicios en modo SaaS o IaaS.

- Dev-Ops: continuando con la línea anterior, no se ha contemplado un sistema CI-CD que permita a los desarrolladores de la aplicación, del e-commerce en este caso, poder trabajar en las mejoras del aplicativo. Un sistema de integración continua como GitLab o Jenkins, usando el repositorio de código correspondiente, podría dotar a los desarrolladores de independencia para poder desplegar, por un lado, instancias de preproducción, test o QA, y solicitar por otro lado la liberación de funcionalidades en producción de forma rápida y sencilla.

7. Glosario

E-Commerce: acrónimo inglés de *electronic commerce*, actividad de compraventa electrónica a través de internet.

Black Friday: día de inauguración de la temporada de compras de navidad, acuñado en Filadelfia E.U.A.

Software: componentes lógicos de permiten la ejecución de tareas informáticas.
Hipervisor: capa software, programa o sistema operativo, que permite virtualizar un hardware con objetivo de ejecutar varios sistemas que lo exploten.

Nube: recursos informáticos en red disponibles en centros de datos a través de internet, normalmente en régimen de pago por uso.

Servidor: software que acepta peticiones y devuelve el resultado de procesar éstas.

Servidor físico: ordenador de propósito especial, configurado para satisfacer una necesidad específica.

Servidor (máquina) virtual: ordenador de propósito especial, configurado para satisfacer una necesidad específica, ejecutándose virtualizado sobre un hipervisor.

Contendor: aplicación o servicio empaquetado listo para ejecutar en un hipervisor, altamente dependiente del mismo.

Nginx: software servidor Web.

Intérpretes (trabajadores): instancias de un programa que únicamente reciben una tarea, la procesan y devuelven el resultado.

PHP: Lenguaje de programación especialmente adaptado al desarrollo web.

PHP-FPM: implementación de interpretación de PHP mediante el uso del protocolo FastCGI (FastCGI Process Manager).

MariaDB: Motor de base de datos con licencia GPL.

MySQL: Motor de base de datos con licencia dual GPL/Comercial.

A.C.I.D.: Acrónimo en inglés de **A**tomicity, **C**onsistency, **I**solation and **D**urability: Atomicidad, Consistencia, Aislamiento y Durabilidad en castellano, características que debe cumplir un software para considerarse transaccional.

SSD: Acrónimo en inglés de Solid-State Drive, disco que usa almacenamiento en memoria no volátil, como la memoria flash, en detrimento de los discos magnéticos tradicionales.

Offloader: elemento software o hardware, que permite descargar la ejecución de una tarea específica al sistema central.

TCP/IP: protocolo de internet.

no-SQL: tipo de software de gestión de bases de datos que no usa un lenguaje estructurado para realizar consultas.

SSL: acrónimo inglés de Secure Socket Layer, una capa de seguridad extra para protocolo TCP/IP en el cual la información viaja cifrada.

SSH: acrónimo del inglés de Secure SHell, un protocolo de acceso a consolas remotas utilizando el estándar SSL.

SGBD: acrónimo de Sistema Gestor de Bases de Datos.

Consola: terminal en modo texto en la que se introducen órdenes en un sistema informático.

Dev-Ops: acrónimo en inglés de DEvelopment OPErations, conjunto de prácticas de software y tecnologías de la información cuyo objetivo es acelerar el ciclo de vida del desarrollo de software.

Frontend: interfaz gráfica de una aplicación, que permite al usuario interactuar con esta.

Backoffice: interfaz gráfica de una aplicación, que permite a los administradores interactuar con esta.

Elasticsearch: software de búsqueda programado en java, publicado bajo licencia Apache.

Varnish: software acelerador de páginas web dinámicas licenciado bajo BSD.

GlusterFS: sistema de archivos distribuido multiescalable.

Clúster: sistema de procesamiento paralelo o distribuido, para equilibrar carga y/o proporcionar tolerancia a fallos y/o alta disponibilidad.

Stack LEMP: acrónimo utilizado para describir un sistema de servicio Web basado en Linux, Nginx (de la pronunciación 'Enguinx'), PHP y MySQL

Stack LAMP: acrónimo utilizado para describir un sistema de servicio Web basado en Linux, Apache, PHP y MySQL

Microservicio: unidad de software independientemente reemplazable y actualizable.

KVM: Kernel-based Virtual Machine o KVM, (en español, Máquina virtual basada en el núcleo) es una solución para implementar virtualización completa con Linux.

LXC: acrónimo de LinuX Containers es una tecnología de virtualización a nivel de sistema operativo para Linux.

Caché: componente de hardware o software que guarda datos para que las solicitudes futuras de esos datos se puedan atender con mayor rapidez.

Full page cache: sistema de caché acelerador de HTML, el cual guarda toda la estructura y objetos de una página entera, de forma que siguientes peticiones se sirvan directamente desde memoria acelerando dramáticamente el proceso.

Router: es un dispositivo que permite interconectar redes.

NAT: mecanismo utilizado por routers IP para cambiar paquetes entre dos redes que asignan mutuamente direcciones incompatibles.

On-premises: característica de un sistema que implica que tanto aplicativo y el software y hardware necesario para su funcionamiento son proporcionados internamente por la organización o individuo propietario del mismo.

PYME: acrónimo de Pequeña Y Mediana Empresa.

LAN: acrónimo en inglés de Local Area Network, i.e. red de área local.

MAC: acrónimo en inglés de Media Access Control, un identificador de 48 bits exclusivo de un dispositivo de red.

DHCP: acrónimo en inglés de Dynamic Host Configuration Protocol, protocolo de red mediante un servidor asigna dinámicamente una dirección IP, máscara, y distintas direcciones de servicios como DNS o la puerta de enlace en una red dada.

DNS: acrónimo en inglés de Domain Name System, un servicio que permite traducir nombres de dispositivos de red en formato amigable en direcciones de red.

SATA: acrónimo de Serial Advanced Technology Attachment, interfaz de bus en ordenadores que permite la transferencia de datos entre dispositivos de almacenamiento y el sistema.

SSD: acrónimo inglés de Solid State Disk, discos de almacenamiento sin partes móviles, basados en memoria no volátil, como la memoria Flash.

RAM: acrónimo en inglés de Random Access Memory, la memoria principal de un sistema informático.

Datastore: anglicismo de almacén de datos.

Backup: anglicismo de copia de seguridad.

PFSense: software basado en FreeBSD para ordenadores que permite usarlos como dispositivos de red (routers).

Linux (GNU/Linux): sistema operativo de software libre y código abierto basado en UNIX creado por Linus Torvalds en 1991.

Debian: es un sistema operativo libre, basado en GNU/Linux, desarrollado por miles de voluntarios de todo el mundo, que colaboran a través de Internet.

Red Hat: es una multinacional estadounidense de software que provee software de código abierto principalmente a empresas.

Open source: es un modelo de desarrollo de software basado en la colaboración abierta.

Sun Microsystems: fue una empresa informática que se dedicaba a vender estaciones de trabajo, servidores, componentes informáticos, software (sistemas operativos) y servicios informáticos.

Oracle Corporation: es una compañía especializada en el desarrollo de soluciones de nube y locales.

BSD: acrónimo en inglés de Berkeley Software Distribution, fue un sistema operativo derivado de Unix que nace a partir de los aportes realizados a ese sistema por la Universidad de California en Berkeley.

Apache: es un servidor web HTTP de código abierto.

API: acrónimo en inglés de Application Programming Interface, interfaz de programación de aplicaciones, conjunto de utilidades de que ofrece cierto software para ser usado por otro como una capa de abstracción.

Restful: acrónimo en inglés de Representational State Transfer un estilo de arquitectura para sistemas hiperdistribuidos como la World Wide Web.

WWW, World Wide Web, o simplemente "La Web": es un sistema que funciona a través de Internet, por el cual se pueden transmitir diversos tipos de datos a través del Protocolo de Transferencia de Hipertextos o HTTP, que son los enlaces de la página web.

JSON: formato de texto para el intercambio de datos.

Gpg: acrónimo en inglés de GNU Privacy Guard, herramienta de cifrado y firma digital licenciado mediante GPL.

OOM: acrónimo inglés de Out Of Memory estado inconsistente de un sistema informático, el cual se ha quedado sin memoria.

JVM: acrónimo inglés de Java Virtual Machine, una interfaz dependiente del hardware que la ejecuta, la cual permite traducir código Java al código máquina correspondiente a dicho hardware.

Offloading: del inglés descargar, o aliviar carga.

Proxy: servidor intermediario ente las peticiones de clientes a servidores en una arquitectura cliente-servidor.

Proxy reverso o inverso: aplicación específica de un servidor proxy en un sistema de alojamiento web, el cual permite habilitar mecanismos extra como cifrado, distribución de carga, y cacheado.

Host: del inglés anfitrión, se usa para referirse a una computadora en red que provee o usa servicios en esta.

8. Bibliografía

La fecha anotada es la del último acceso a la página.

<https://en.wikipedia.org> (Recurrente, +100 visitas)

<https://es.wikipedia.org> (Recurrente, +100 visitas)

<https://experienceleague.adobe.com/docs/commerce-operations/installation-guide/overview.html?lang=es> (Recurrente, +50 visitas)

<https://www.osradar.com/install-and-configure-glusterfs-debian-10/> (12/11/2022)

<https://sysadmins.co.za/adding-bricks-to-your-glusterfs-volume-to-increase-storage-capacity/> (12/11/2022)

<https://sysadmins.co.za/setup-a-3-node-replicated-storage-volume-with-glusterfs/> (12/11/2022)

<https://docs.redis.com/latest/rs/installing-upgrading/supported-platforms/> (19/11/2022)

<https://docs.redis.com/latest/rs/installing-upgrading/> (19/11/2022)

<https://nginx.org/en/docs/> (27/11/2022)

<https://www.nginx.com/resources/wiki/start/topics/examples/full/> (27/11/2022)

https://nginx.org/en/docs/http/configuring_https_servers.html (27/11/2022)

<https://mariadb.org/download/?t=repo-config&d=Debian+10+%22buster%22> (27/11/2022)

<https://www.elastic.co/guide/en/Elasticsearch/reference/7.17/deb.html> (3/12/2022)

<https://medium.com/trendyol-tech/how-to-configure-Elasticsearch-heap-size-to-change-max-memory-size-cb9ca016ce06> (3/12/2022)

<https://www.elastic.co/guide/en/Elasticsearch/reference/7.17/system-config.html> (3/12/2022)

<https://www.elastic.co/guide/en/Elasticsearch/reference/7.17/advanced-configuration.html#set-jvm-heap-size> (3/12/2022)

<https://howtoforge.com/how-to-install-php-8-on-debian-11/> (7/12/2022)

<https://www.tecmint.com/mariadb-master-slave-replication-on-centos-rhel-debian/> (18/12/2022)

<http://woshub.com/configure-mariadb-replication/> (18/12/2022)

<https://www.howtoforge.com/how-to-install-and-configure-redis-6-on-debian-11/>
(19/12/2022)

<https://redis.io/docs/management/scaling/> (19/12/2022)

<https://www.digitalocean.com/community/tutorials/how-to-configure-a-redis-cluster-on-centos-7> (19/12/2022)

<https://www.thegeekstuff.com/2019/05/install-Elasticsearch-cluster/>
(20/12/2022)

<https://medium.com/codeops/how-to-setup-an-Elasticsearch-cluster-55a057d60d6c> (20/12/2022)

<https://techoverflow.net/2020/12/16/how-to-fix-Elasticsearch-1-initial-heap-size-not-equal-to-maximum-heap-size/> (20/12/2022)

9. Anexos

9.1 Sistema de copias de seguridad.

El sistema de copia de seguridad queda, a priori, fuera de alcance de este trabajo por la limitación en el hardware disponible para llevar a cabo el proyecto. Sin embargo, una política de copias de seguridad coherente es necesaria para todo tipo de proyecto, sea personal o profesional, por lo que se ha decidido incluir este pequeño anexo para poder ofrecer una imagen simple de qué, cómo y cuándo se copian, tanto las máquinas necesarias como la documentación de este trabajo.

Se ha utilizado un servidor preexistente basado en Proxmox Backup Server, creado por la misma organización que Proxmox Virtual Environment, por lo que son dos entornos totalmente integrados y de gran facilidad de uso y administración.

Una copia de todas las máquinas y contenedores de este trabajo estará disponible para descargar si así lo solicita el tribunal de proyectos.

En el servidor preexistente de Proxmox Backup Server se ha añadido un Datastore para guardar las copias de todas las máquinas y contenedores de la infraestructura:

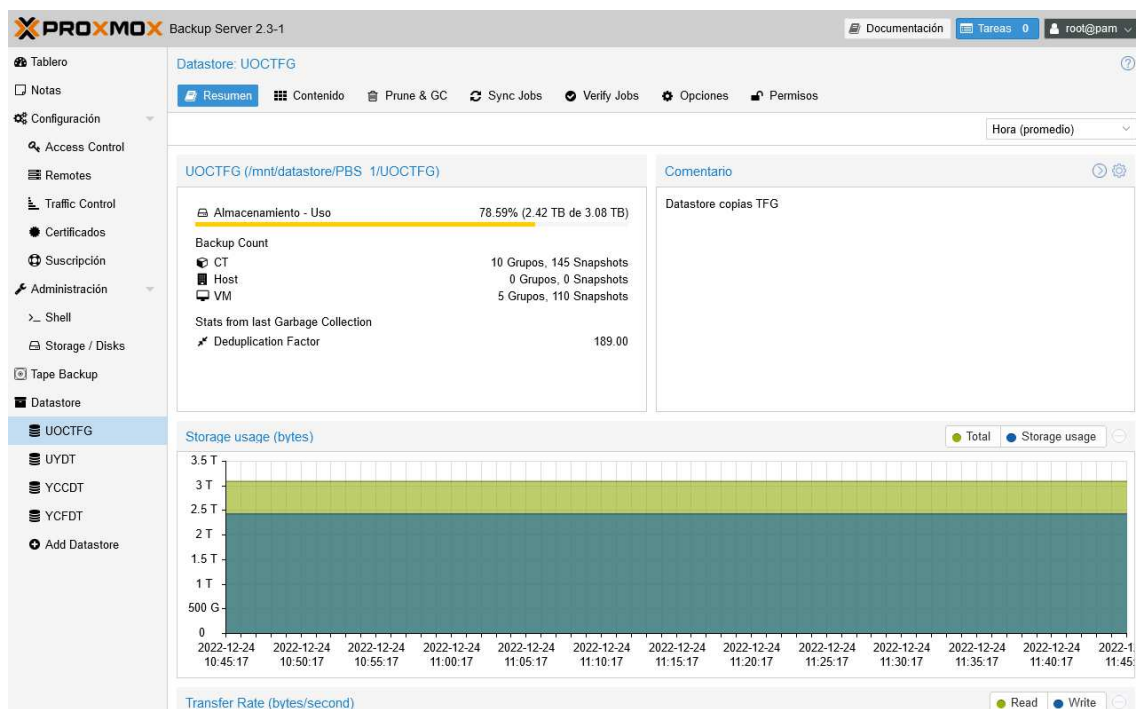


Ilustración 104 - Datastore copias de seguridad.

En el servidor que ejecuta las máquinas de la infraestructura se añade un almacenamiento del tipo Proxmox Backup Server y se configuran dos tareas para hacer las copias de seguridad.

Han de ser dos tareas separadas debido a que los contenedores que montan el sistema de archivos compartido por GlusterFS no son compatibles con la copia de seguridad mediante *snapshot*, por limitaciones del tipo de contenedor LXC, por lo que se han de hacer mediante apagado y encendido de los mismos, estos contenedores son concretamente los balanceadores y los intérpretes de PHP:

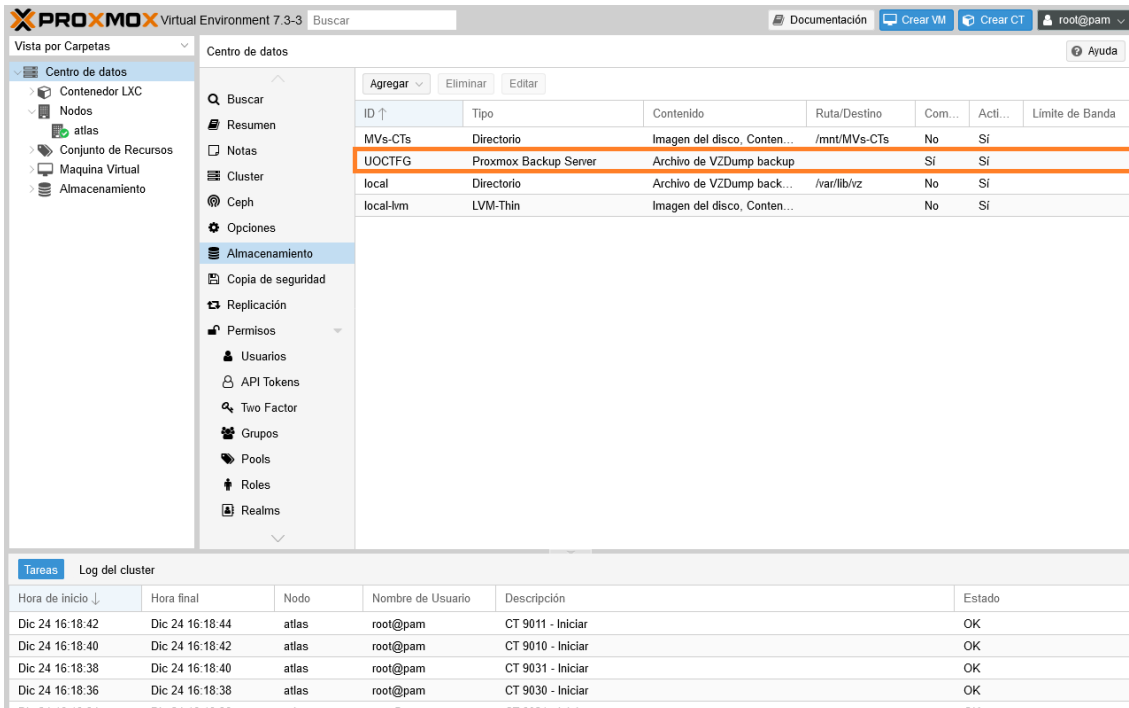


Ilustración 105 - Configuración almacenamiento copias de seguridad.

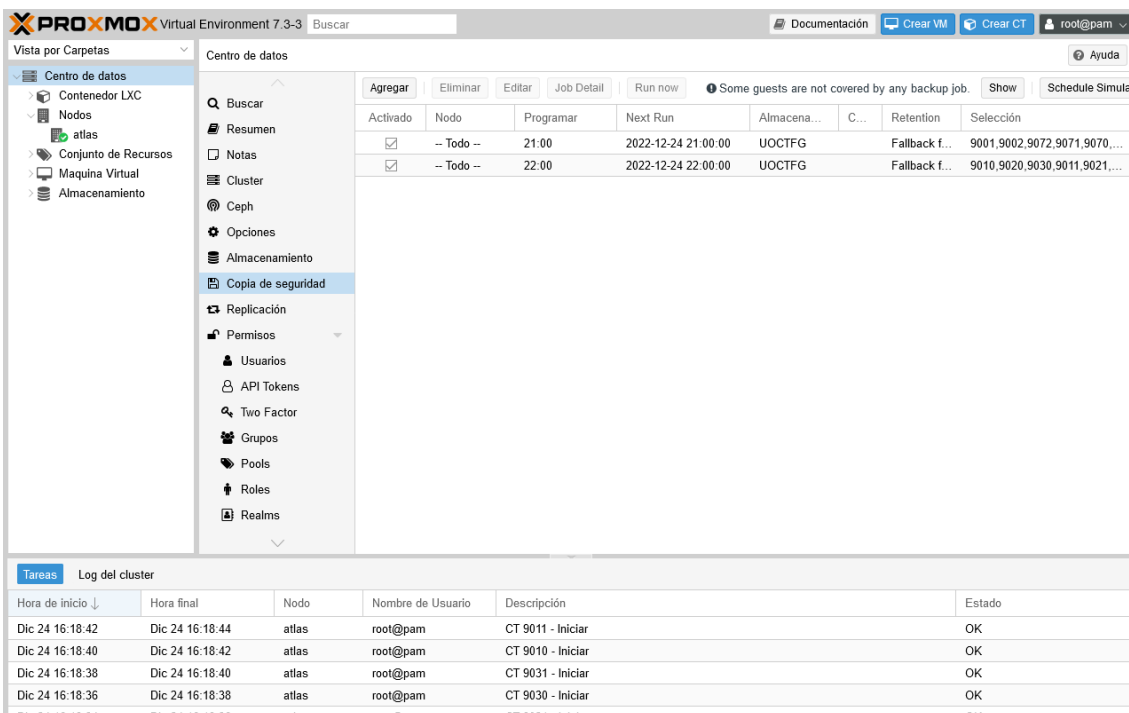


Ilustración 106 - Programación de copias de seguridad.

En cuanto al directorio de trabajo, que contiene esta memoria, así como otros archivos necesarios para la gestión día a día, se guarda en un servidor personal en la nube funcionando sobre software Nextcloud en un stack LAMP. El directorio se sincroniza constantemente con dicho servidor en la nube, y éste, que es una máquina virtual, se le hace copia de seguridad con programación diaria. Además, existe una copia de los datos internos del propio servidor a otro dispositivo para cumplir la regla del 3-2-1.

9.2. Listado de IP:

A continuación, el listado de todas las IP de red locales que se han configurado en las máquinas que conforman la infraestructura:

IP	Nombre Host	Notas
10.20.22.1	Router-PFS	Máquina virtual Router PFSense
10.20.22.2	Control-D11	Máquina virtual Control
10.20.22.10	LB1-D11	Contenedor Load Balancer y full page cache 1
10.20.22.11	LB2-D11	Contenedor Load Balancer y full page cache 2
10.20.22.20	FE1-D11	Contenedor <i>Frontend</i> 1
10.20.22.21	FE2-D11	Contenedor <i>Frontend</i> 2
10.20.22.30	BE1-D11	Contenedor <i>Backend</i> 1
10.20.22.31	BE2-D11	Contenedor <i>Backend</i> 2
10.20.22.40	SGBD1-D11	Contenedor MySQL principal 1
10.20.22.41	SGBD2-D11	Contenedor MySQL secundario
10.20.22.50	RDS1-D11	Contenedor Redis 1
10.20.22.51	RDS2-D11	Contenedor Redis 2
10.20.22.60	ELS1-D11	Contenedor Elasticsearch 1
10.20.22.61	ELS2-D11	Contenedor Elasticsearch 2
10.20.22.62	ELS3-D11	Contenedor Elasticsearch 3
10.20.22.70	STR1-D11	Máquina virtual Almacenamiento GlusterFS 1
10.20.22.71	STR2-D11	Máquina virtual Almacenamiento GlusterFS 2
10.20.22.72	STR3-D11	Máquina virtual Almacenamiento GlusterFS 3

9.3. Archivos de configuración modificados para GlusterFS:

/etc/hosts (todos los nodos)

```
127.0.0.1    localhost
127.0.1.1    STRX-D11.tfg.local    STRX-D11

# The following lines are desirable for IPv6 capable hosts
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters

10.20.22.70 str1 STR1-D11
10.20.22.71 str2 STR2-D11
10.20.22.72 str3 STR3-D11
```

/etc/fstab STR1-D11:

```

# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# systemd generates mount units based on this file, see systemd.mount(5).
# Please run 'systemctl daemon-reload' after making changes here.
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=e78db844-617a-4959-82cd-cabfdada4341 / ext4 errors=remount-ro 0
1
# swap was on /dev/sda2 during installation
UUID=3fabec51-99dc-48f9-8f33-75ea8bb7cdd3 none swap sw 0
0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
UUID=360f5c4c-d86d-4ec7-a537-5bb596538a76 /gluster/bricks/1 ext4 defaults 0 1

```

/etc/fstab STR2-D11:

```

# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# systemd generates mount units based on this file, see systemd.mount(5).
# Please run 'systemctl daemon-reload' after making changes here.
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=e78db844-617a-4959-82cd-cabfdada4341 / ext4 errors=remount-ro 0
1
# swap was on /dev/sda2 during installation
UUID=3fabec51-99dc-48f9-8f33-75ea8bb7cdd3 none swap sw 0
0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
UUID=9dd538c9-85be-4149-b1c6-9ad71c722259 /gluster/bricks/2 ext4 defaults 0 1

```

/etc/fstab STR3-D11:

```

# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# systemd generates mount units based on this file, see systemd.mount(5).
# Please run 'systemctl daemon-reload' after making changes here.
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=e78db844-617a-4959-82cd-cabfdada4341 / ext4 errors=remount-ro 0
1
# swap was on /dev/sda2 during installation
UUID=3fabec51-99dc-48f9-8f33-75ea8bb7cdd3 none swap sw 0
0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
UUID=184b9e4f-fc5c-4fe4-a2f3-ae1d3d98b063 /gluster/bricks/3 ext4 defaults 0 1

```

9.4. Archivos de configuración modificados para PHP.

Todos los intérpretes:

/etc/fstab:

```

# UNCONFIGURED FSTAB FOR BASE SYSTEM
str:/ecommerce /media/ecommerce glusterfs defaults,_netdev 0 0

```

/etc/php/8.1/fpm/pool.d/www.conf, se omiten las líneas comentadas:

```
[www]
user = www-data
group = www-data
listen = 9000
listen.owner = www-data
listen.group = www-data
pm = dynamic
pm.max_children = 5
pm.start_servers = 2
pm.min_spare_servers = 1
pm.max_spare_servers = 3
```

9.5. Archivos de configuración modificados para MariaDB.

/etc/mysql/my.cnf, servidor activo, se excluyen las líneas comentadas:

```
[client]
port                = 3306
socket              = /var/run/mysqld/mysqld.sock

[mysqld_safe]
socket              = /var/run/mysqld/mysqld.sock
nice                = 0

[mysqld]
user                = mysql
pid-file            = /var/run/mysqld/mysqld.pid
socket              = /var/run/mysqld/mysqld.sock
port                = 3306
basedir             = /usr
datadir             = /var/lib/mysql
tmpdir              = /tmp
lc_messages_dir    = /usr/share/mysql
lc_messages         = en_US
skip-external-locking
bind-address        = 0.0.0.0
max_connections     = 100
connect_timeout     = 5
wait_timeout        = 600
max_allowed_packet = 16M
thread_cache_size   = 128
sort_buffer_size    = 4M
bulk_insert_buffer_size = 16M
tmp_table_size      = 32M
max_heap_table_size = 32M
myisam_recover_options = BACKUP
key_buffer_size     = 128M
table_open_cache    = 400
myisam_sort_buffer_size = 512M
concurrent_insert   = 2
read_buffer_size    = 2M
read_rnd_buffer_size = 1M
query_cache_limit   = 128K
query_cache_size    = 64M
log_warnings        = 2
slow_query_log_file = /var/log/mysql/mariadb-slow.log
long_query_time     = 10
log_slow_verbosity  = query_plan

server-id           = 1
report_host         = SGBD1
log_bin             = /var/log/mysql/mariadb-bin
log_bin_index       = /var/log/mysql/mariadb-bin.index
expire_logs_days    = 10
max_binlog_size     = 100M
relay_log           = /var/log/mysql/relay-bin
relay_log_index     = /var/log/mysql/relay-bin.index
default_storage_engine = InnoDB
innodb_buffer_pool_size = 256M
```

```
innodb_log_buffer_size = 8M
innodb_file_per_table = 1
innodb_open_files = 400
innodb_io_capacity = 400
innodb_flush_method = O_DIRECT
```

[galera]

```
[mysqldump]
quick
quote-names
max_allowed_packet = 16M
```

/etc/mysql/my.cnf, servidor pasivo, se excluyen las líneas comentadas:

```
[client]
port = 3306
socket = /var/run/mysqld/mysqld.sock
```

```
[mysqld_safe]
socket = /var/run/mysqld/mysqld.sock
nice = 0
```

```
[mysqld]
user = mysql
pid-file = /var/run/mysqld/mysqld.pid
socket = /var/run/mysqld/mysqld.sock
port = 3306
basedir = /usr
datadir = /var/lib/mysql
tmpdir = /tmp
lc_messages_dir = /usr/share/mysql
lc_messages = en_US
skip-external-locking
bind-address = 0.0.0.0
max_connections = 100
connect_timeout = 5
wait_timeout = 600
max_allowed_packet = 16M
thread_cache_size = 128
sort_buffer_size = 4M
bulk_insert_buffer_size = 16M
tmp_table_size = 32M
max_heap_table_size = 32M
myisam_recover_options = BACKUP
key_buffer_size = 128M
table_open_cache = 400
myisam_sort_buffer_size = 512M
concurrent_insert = 2
read_buffer_size = 2M
read_rnd_buffer_size = 1M
query_cache_limit = 128K
query_cache_size = 64M
log_warnings = 2
slow_query_log_file = /var/log/mysql/mariadb-slow.log
long_query_time = 10
log_slow_verbosity = query_plan

server-id = 2
report_host = SGBD2
log_bin = /var/log/mysql/mariadb-bin
log_bin_index = /var/log/mysql/mariadb-bin.index
expire_logs_days = 10
max_binlog_size = 100M
relay_log = /var/log/mysql/relay-bin
relay_log_index = /var/log/mysql/relay-bin.index
default_storage_engine = InnoDB
innodb_buffer_pool_size = 256M
innodb_log_buffer_size = 8M
innodb_file_per_table = 1
innodb_open_files = 400
innodb_io_capacity = 400
```

```
innodb_flush_method = O_DIRECT
```

```
[galera]
```

```
[mysqldump]
```

```
quick
```

```
quote-names
```

```
max_allowed_packet = 16M
```

```
[mysql]
```

```
[isamchk]
```

```
key_buffer = 16M
```

9.6. Archivos de configuración modificados para Redis.

/etc/redis/redis.conf, servidor principal, se excluyen las líneas comentadas:

```
protected-mode no
port 6379
tcp-backlog 511
timeout 0
tcp-keepalive 300
daemonize yes
supervised auto
pidfile /run/redis/redis-server.pid
loglevel notice
logfile /var/log/redis/redis-server.log
databases 16
always-show-logo no
set-proc-title yes
proc-title-template "{title} {listen-addr} {server-mode}"
stop-writes-on-bgsave-error yes
rdbcompression yes
rdbchecksum yes
dbfilename dump.rdb
rdb-del-sync-files no
dir /var/lib/redis
replica-serve-stale-data yes
replica-read-only yes
repl-diskless-sync yes
repl-diskless-sync-delay 5
repl-diskless-sync-max-replicas 0
repl-diskless-load disabled
repl-disable-tcp-nodelay no
replica-priority 100
acllog-max-len 128
lazyfree-lazy-eviction no
lazyfree-lazy-expire no
lazyfree-lazy-server-del no
replica-lazy-flush no
lazyfree-lazy-user-del no
lazyfree-lazy-user-flush no
oom-score-adj no
oom-score-adj-values 0 200 800
disable-thp yes
appendonly no
appendfilename "appendonly.aof"
appenddirname "appendonlydir"
appendfsync everysec
no-appendfsync-on-rewrite no
auto-aof-rewrite-percentage 100
auto-aof-rewrite-min-size 64mb
aof-load-truncated yes
aof-use-rdb-preamble yes
aof-timestamp-enabled no
slowlog-log-slower-than 10000
slowlog-max-len 128
latency-monitor-threshold 0
notify-keyspace-events ""
hash-max-listpack-entries 512
hash-max-listpack-value 64
```



```

list-max-listpack-size -2
list-compress-depth 0
set-max-intset-entries 512
zset-max-listpack-entries 128
zset-max-listpack-value 64
hll-sparse-max-bytes 3000
stream-node-max-bytes 4096
stream-node-max-entries 100
activerehashing yes
client-output-buffer-limit normal 0 0 0
client-output-buffer-limit replica 256mb 64mb 60
client-output-buffer-limit pubsub 32mb 8mb 60
hz 10
dynamic-hz yes
aof-rewrite-incremental-fsync yes
rdb-save-incremental-fsync yes
jemalloc-bg-thread yes

```

/etc/redis/redis.conf, servidor secundario, se excluyen las líneas comentadas:

```

protected-mode no
port 6379
tcp-backlog 511
timeout 0
tcp-keepalive 300
daemonize yes
supervised auto
pidfile /run/redis/redis-server.pid
loglevel notice
logfile /var/log/redis/redis-server.log
databases 16
always-show-logo no
set-proc-title yes
proc-title-template "{title} {listen-addr} {server-mode}"
stop-writes-on-bgsave-error yes
rdbcompression yes
rdbchecksum yes
dbfilename dump.rdb
rdb-del-sync-files no
dir /var/lib/redis
replica-serve-stale-data yes
replica-read-only yes
repl-diskless-sync yes
repl-diskless-sync-delay 5
repl-diskless-sync-max-replicas 0
repl-diskless-load disabled
repl-disable-tcp-nodelay no
replica-priority 100
acllog-max-len 128
lazyfree-lazy-eviction no
lazyfree-lazy-expire no
lazyfree-lazy-server-del no
replica-lazy-flush no
lazyfree-lazy-user-del no
lazyfree-lazy-user-flush no
oom-score-adj no
oom-score-adj-values 0 200 800
disable-thp yes
appendonly no
appendfilename "appendonly.aof"
appenddirname "appendonlydir"
appendfsync everysec
no-appendfsync-on-rewrite no
auto-aof-rewrite-percentage 100
auto-aof-rewrite-min-size 64mb
aof-load-truncated yes
aof-use-rdb-preamble yes
aof-timestamp-enabled no
slowlog-log-slower-than 10000
slowlog-max-len 128
latency-monitor-threshold 0
notify-keyspace-events ""
hash-max-listpack-entries 512
hash-max-listpack-value 64

```

```

list-max-listpack-size -2
list-compress-depth 0
set-max-intset-entries 512
zset-max-listpack-entries 128
zset-max-listpack-value 64
hll-sparse-max-bytes 3000
stream-node-max-bytes 4096
stream-node-max-entries 100
activerehashing yes
client-output-buffer-limit normal 0 0 0
client-output-buffer-limit replica 256mb 64mb 60
client-output-buffer-limit pubsub 32mb 8mb 60
hz 10
dynamic-hz yes
aof-rewrite-incremental-fsync yes
rdb-save-incremental-fsync yes
jemalloc-bg-thread yes
slaveof 10.20.22.50 6379

```

9.7. Archivos de configuración modificados para Elasticsearch.

/etc/Elasticsearch/jvm.options, todos los containers, se excluyen las líneas comentadas:

```

-Xms3g
-Xmx3g
8-13:-XX:+UseConcMarkSweepGC
8-13:-XX:CMSInitiatingOccupancyFraction=75
8-13:-XX:+UseCMSInitiatingOccupancyOnly
14-:-XX:+UseG1GC
-Djava.io.tmpdir=${ES_TMPDIR}
-XX:+HeapDumpOnOutOfMemoryError
9-:-XX:+ExitOnOutOfMemoryError
-XX:HeapDumpPath=/var/lib/Elasticsearch
-XX:ErrorFile=/var/log/Elasticsearch/hs_err_pid%p.log
8:-XX:+PrintGCDetails
8:-XX:+PrintGCDateStamps
8:-XX:+PrintTenuringDistribution
8:-XX:+PrintGCApplicationStoppedTime
8:-Xloggc:/var/log/Elasticsearch/gc.log
8:-XX:+UseGCLogFileRotation
8:-XX:NumberOfGCLogFiles=32
8:-XX:GCLogFileSize=64m
9-:-
Xlog:gc*,gc+age=trace,safepoint:file=/var/log/Elasticsearch/gc.log:utctime,pid,tags:filecount=32,filesize=64m

```

/etc/Elasticsearch/Elasticsearch.yml, els1, se excluyen las líneas comentadas:

```

cluster.name: els-magento
node.master: true
node.data: true
node.name: els1
path.data: /var/lib/Elasticsearch
path.logs: /var/log/Elasticsearch
network.host: 0.0.0.0
http.port: 9200
discovery.seed_hosts: ["els1", "els2", "els3"]
cluster.initial_master_nodes: ["els1", "els2"]

```

/etc/Elasticsearch/Elasticsearch.yml, els2, se excluyen las líneas comentadas:

```

cluster.name: els-magento
node.master: true
node.data: true
node.name: els2
path.data: /var/lib/Elasticsearch
path.logs: /var/log/Elasticsearch
network.host: 0.0.0.0

```

```

http.port: 9200
discovery.seed_hosts: ["els1", "els2", "els3"]
cluster.initial_master_nodes: ["els1", "els2"]

```

/etc/Elasticsearch/Elasticsearch.yml, els3, se excluyen las líneas comentadas:

```

cluster.name: els-magento
node.master: true
node.data: true
node.name: els3
path.data: /var/lib/Elasticsearch
path.logs: /var/log/Elasticsearch
network.host: 0.0.0.0
http.port: 9200
discovery.seed_hosts: ["els1", "els2", "els3"]
cluster.initial_master_nodes: ["els1", "els2"]

```

9.7. Archivos de configuración modificados para Varnish.

/etc/varnish/default.vcl

```

# VCL version 5.0 is not supported so it should be 4.0 even though actually used Varnish
version is 6
vcl 4.0;

import std;
# The minimal Varnish version is 6.0
# For SSL offloading , pass the following header in your proxy server or load balancer:
'X-Forwarded-Proto: https'

backend default {
    .host = "localhost";
    .port = "8080";
    .first_byte_timeout = 600s;
    #.probe = {
    #   .url = "/pub/health_check.php";
    #   .timeout = 2s;
    #   .interval = 5s;
    #   .window = 10;
    #   .threshold = 5;
    #}
}

acl purge {
    "10.20.22.20";
    "10.20.22.30";
    "10.20.22.2";
}

sub vcl_recv {
    if (req.restarts > 0) {
        set req.hash_always_miss = true;
    }

    if (req.method == "PURGE") {
        if (client.ip !~ purge) {
            return (synth(405, "Method not allowed"));
        }
        # To use the X-Pool header for purging varnish during automated deployments, make
sure the X-Pool header
        # has been added to the response in your backend server config. This is used, for
example, by the
        # capistrano-magento2 gem for purging old content from varnish during it's deploy
routine.
        if (!req.http.X-Magento-Tags-Pattern && !req.http.X-Pool) {
            return (synth(400, "X-Magento-Tags-Pattern or X-Pool header required"));
        }
        if (req.http.X-Magento-Tags-Pattern) {
            ban("obj.http.X-Magento-Tags ~ " + req.http.X-Magento-Tags-Pattern);
        }
    }
}

```

```

    if (req.http.X-Pool) {
        ban("obj.http.X-Pool ~ " + req.http.X-Pool);
    }
    return (synth(200, "Purged"));
}

if (req.method != "GET" &&
    req.method != "HEAD" &&
    req.method != "PUT" &&
    req.method != "POST" &&
    req.method != "TRACE" &&
    req.method != "OPTIONS" &&
    req.method != "DELETE") {
    /* Non-RFC2616 or CONNECT which is weird. */
    return (pipe);
}

# We only deal with GET and HEAD by default
if (req.method != "GET" && req.method != "HEAD") {
    return (pass);
}

# Bypass customer, shopping cart, checkout
if (req.url ~ "/customer" || req.url ~ "/checkout") {
    return (pass);
}

# Bypass health check requests
if (req.url ~ "^(pub/)?(health_check.php)$") {
    return (pass);
}

# Set initial grace period usage status
set req.http.grace = "none";

# normalize url in case of leading HTTP scheme and domain
set req.url = regsub(req.url, "^http[s]?://", "");

# collect all cookies
std.collect(req.http.Cookie);

# Compression filter. See https://www.varnish-cache.org/trac/wiki/FAQ/Compression
if (req.http.Accept-Encoding) {
    if (req.url ~ "\.(jpg|jpeg|png|gif|gz|tgz|bz2|tbz|mp3|ogg|swf|flv)$") {
        # No point in compressing these
        unset req.http.Accept-Encoding;
    } elseif (req.http.Accept-Encoding ~ "gzip") {
        set req.http.Accept-Encoding = "gzip";
    } elseif (req.http.Accept-Encoding ~ "deflate" && req.http.user-agent !~ "MSIE") {
        set req.http.Accept-Encoding = "deflate";
    } else {
        # unknown algorithm
        unset req.http.Accept-Encoding;
    }
}

# Remove all marketing get parameters to minimize the cache objects
if (req.url ~ "(?|&)(clid|cx|ie|cof|siteurl|zanpid|origin|fbclid|mc_[a-z]+|utm_[a-z]+|_bta_[a-z]+)=") {
    set req.url = regsuball(req.url, "(clid|cx|ie|cof|siteurl|zanpid|origin|fbclid|mc_[a-z]+|utm_[a-z]+|_bta_[a-z]+)=[-_A-z0-9+()%.]+&?", "");
    set req.url = regsub(req.url, "[?|&]+$", "");
}

# Static files caching
if (req.url ~ "^(pub/)?(media|static)/") {
    # Static files should not be cached by default
    return (pass);

    # But if you use a few locales and don't use CDN you can enable caching static
    # files by commenting previous line (#return (pass);) and uncommenting next 3 lines
    #unset req.http.Https;
    #unset req.http.X-Forwarded-Proto;
    #unset req.http.Cookie;
}

```

```

    }

    # Bypass authenticated GraphQL requests without a X-Magento-Cache-Id
    if (req.url ~ "/graphql" && !req.http.X-Magento-Cache-Id && req.http.Authorization ~
    "Bearer") {
        return (pass);
    }

    return (hash);
}

sub vcl_hash {
    if ((req.url !~ "/graphql" || !req.http.X-Magento-Cache-Id) && req.http.cookie ~ "X-
    Magento-Vary=") {
        hash_data(regsub(req.http.cookie, "^.*X-Magento-Vary=([^;]+);*.*$", "\1"));
    }

    # To make sure http users don't see ssl warning
    if (req.http.X-Forwarded-Proto) {
        hash_data(req.http.X-Forwarded-Proto);
    }

    if (req.url ~ "/graphql") {
        call process_graphql_headers;
    }
}

sub process_graphql_headers {
    if (req.http.X-Magento-Cache-Id) {
        hash_data(req.http.X-Magento-Cache-Id);

        # When the frontend stops sending the auth token, make sure users stop getting
        results cached for logged-in users
        if (req.http.Authorization ~ "Bearer") {
            hash_data("Authorized");
        }
    }

    if (req.http.Store) {
        hash_data(req.http.Store);
    }

    if (req.http.Content-Currency) {
        hash_data(req.http.Content-Currency);
    }
}

sub vcl_backend_response {

    set beresp.grace = 3d;

    if (beresp.http.content-type ~ "text") {
        set beresp.do_esf = true;
    }

    if (bereq.url ~ "\.js$" || beresp.http.content-type ~ "text") {
        set beresp.do_gzip = true;
    }

    if (beresp.http.X-Magento-Debug) {
        set beresp.http.X-Magento-Cache-Control = beresp.http.Cache-Control;
    }

    # cache only successfully responses and 404s that are not marked as private
    if (beresp.status != 200 &&
        beresp.status != 404 &&
        beresp.http.Cache-Control ~ "private") {
        set beresp.uncacheable = true;
        set beresp.ttl = 86400s;
        return (deliver);
    }

    # validate if we need to cache it and prevent from setting cookie
    if (beresp.ttl > 0s && (bereq.method == "GET" || bereq.method == "HEAD")) {

```

```

        unset beresp.http.set-cookie;
    }

    # If page is not cacheable then bypass varnish for 2 minutes as Hit-For-Pass
    if (beresp.ttl <= 0s ||
        beresp.http.Surrogate-control ~ "no-store" ||
        (!beresp.http.Surrogate-Control &&
        beresp.http.Cache-Control ~ "no-cache|no-store") ||
        beresp.http.Vary == "*") {
        # Mark as Hit-For-Pass for the next 2 minutes
        set beresp.ttl = 120s;
        set beresp.uncacheable = true;
    }

    # If the cache key in the Magento response doesn't match the one that was sent in the
    request, don't cache under the request's key
    if (bereg.url ~ "/graphql" && bereq.http.X-Magento-Cache-Id && bereq.http.X-Magento-
    Cache-Id != beresp.http.X-Magento-Cache-Id) {
        set beresp.ttl = 0s;
        set beresp.uncacheable = true;
    }

    return (deliver);
}

sub vcl_deliver {
    if (resp.http.x-varnish ~ " ") {
        set resp.http.X-Magento-Cache-Debug = "HIT";
        set resp.http.Grace = req.http.grace;
    } else {
        set resp.http.X-Magento-Cache-Debug = "MISS";
    }

    # Not letting browser to cache non-static files.
    if (resp.http.Cache-Control !~ "private" && req.url !~ "^/(pub/)?(media|static)/") {
        set resp.http.Pragma = "no-cache";
        set resp.http.Expires = "-1";
        set resp.http.Cache-Control = "no-store, no-cache, must-revalidate, max-age=0";
    }

    if (!resp.http.X-Magento-Debug) {
        unset resp.http.Age;
    }
    unset resp.http.X-Magento-Debug;
    unset resp.http.X-Magento-Tags;
    unset resp.http.X-Powered-By;
    unset resp.http.Server;
    unset resp.http.X-Varnish;
    unset resp.http.Via;
    unset resp.http.Link;
}

sub vcl_hit {
    if (obj.ttl >= 0s) {
        # Hit within TTL period
        return (deliver);
    }
    if (std.healthy(req.backend_hint)) {
        if (obj.ttl + 300s > 0s) {
            # Hit after TTL expiration, but within grace period
            set req.http.grace = "normal (healthy server)";
            return (deliver);
        } else {
            # Hit after TTL and grace expiration
            return (restart);
        }
    }
    } else {
        # server is not healthy, retrieve from cache
        set req.http.grace = "unlimited (unhealthy server)";
        return (deliver);
    }
}
}

```

9.9. Archivos de configuración modificados y creados para Nginx

/etc/nginx/proxy_public.conf

```
#Comportamiento de FRONTEND
### force timeouts if one of backend is died ##
proxy_next_upstream error timeout invalid_header http_500 http_502 http_503;

proxy_connect_timeout 600;
proxy_read_timeout 600;
proxy_send_timeout 600;

#### Set headers ####
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

### Most PHP, Python, Rails, Java App can use this header ###
proxy_set_header X-Forwarded-Proto https;

### By default we don't want to redirect it ####
proxy_redirect off;
```

/etc/nginx/proxy_admin.conf

```
#Comportamiento de BACKOFFICE
#Timeouts
proxy_connect_timeout 1200;
proxy_read_timeout 1200;
proxy_send_timeout 1200;

### force timeouts if one of backend is died ##
proxy_next_upstream error timeout invalid_header http_500 http_502 http_503;

#### Set headers ####
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

### Most PHP, Python, Rails, Java App can use this header ###
proxy_set_header X-Forwarded-Proto $scheme;

### By default we don't want to redirect it ####
proxy_redirect off;
```

/etc/nginx/sites_enabled/int-magento.conf

```
# Intérpretes para front
upstream frontend {
    server 10.20.22.20:9000;
    server 10.20.22.21:9000;
}
# Intérpretes para back
upstream backend {
    server 10.20.22.30:9000;
    server 10.20.22.31:9000;
}

#Map Intérpretes
map $uri $magento{
    default frontend;
    ~/admin backend;
}

map $http_host $MAGE_RUN_CODE {
    default '';
}
# Recoger peticiones una vez han atravesado Varnish a PHP-FPM
server {
```

```

listen 127.0.0.1:8080;
access_log /media/ecommerce/logs/logs_nginx/int-magento.ulla-yanez.local-
access.log;
error_log /media/ecommerce/logs/logs_nginx/int-magento.ulla-yanez.local-error.log;
server_name magento.ulla-yanez.local;
#Algunas variables
set $MAGE_KEEDS_ROOT /media/ecommerce/current/;
root $MAGE_KEEDS_ROOT/pub;
set $MAGE_RUN_TYPE store;
index index.php;
autoindex off;
charset UTF-8;
error_page 404 403 = /errors/404.php; #OJO CAMBIAR!!!!
#add_header "X-UA-Compatible" "IE=Edge";
# Deny archivos sensibles
location /.user.ini {
    deny all;
}

# PHP entry point for setup application
location ~* ^/setup($|/) {
    deny all;
}

# PHP entry point for update application
location ~* ^/update($|/) {
    deny all;
}

location / {
    try_files $uri $uri/ /index.php$is_args$args;
}

location /admin_lqw7wb {
    include proxy_admin.conf;
    try_files $uri $uri/ /index.php?$args;
    set $magento backend;
}

location /pub/ {
    location
^/pub/media/(downloadable|customer|import|theme_customization/.*\.xml) {
        deny all;
    }
    alias $MAGE_KEEDS_ROOT/pub/;
    add_header X-Frame-Options "SAMEORIGIN";
}

location /static/ {
    # Uncomment the following line in production mode
    # expires max;

    # Remove signature of the static files that is used to overcome the
browser cache
    location ~ ^/static/version {
        rewrite ^/static/(version[^/]+)/?(.*)$ /static/$2 last;
    }

    location
\.(ico|jpg|jpeg|png|gif|svg|js|css|swf|eot|ttf|otf|woff|woff2|json)$ {
        add_header Cache-Control "public";
        add_header X-Frame-Options "SAMEORIGIN";
        expires +1y;

        if (!-f $request_filename) {
            rewrite ^/static/?(.*)$ /static.php?resource=$1 last;
        }
    }
    location ~* \.(zip|gz|gzip|bz2|csv|xml)$ {
        add_header Cache-Control "no-store";
        add_header X-Frame-Options "SAMEORIGIN";
        expires off;

        if (!-f $request_filename) {
            rewrite ^/static/?(.*)$ /static.php?resource=$1 last;
        }
    }
}

```



```

    }
    }
    if (!-f $request_filename) {
        rewrite ^/static/?(.*)$ /static.php?resource=$1 last;
    }
    add_header X-Frame-Options "SAMEORIGIN";
}

location /media/ {
    try_files $uri $uri/ /get.php$is_args$args;

    location ~ ^/media/theme_customization/.*\.xml {
        deny all;
    }

    location
\.(ico|jpg|jpeg|png|gif|svg|js|css|swf|eot|ttf|otf|woff|woff2)$ {
        add_header Cache-Control "public";
        add_header X-Frame-Options "SAMEORIGIN";
        expires +1y;
        try_files $uri $uri/ /get.php$is_args$args;
    }
    location ~* \.(zip|gz|gzip|bz2|csv|xml)$ {
        add_header Cache-Control "no-store";
        add_header X-Frame-Options "SAMEORIGIN";
        expires off;
        try_files $uri $uri/ /get.php$is_args$args;
    }
    add_header X-Frame-Options "SAMEORIGIN";
}

location /media/customer/ {
    deny all;
}

location /media/downloadable/ {
    deny all;
}

location /media/import/ {
    deny all;
}

# PHP Aplicacion principal
location ~ (index|get|static|report|404|503)\.php$ {
    try_files $uri =404;
    fastcgi_pass $magento;
    #fastcgi_buffers 1024 4k;
    fastcgi_buffers 256 1024k;
    fastcgi_buffer_size 1024k;
    fastcgi_busy_buffers_size 1024k;

    fastcgi_param HTTPS on;
    fastcgi_read_timeout 600s;
    fastcgi_connect_timeout 600s;
    fastcgi_index index.php;
    fastcgi_param MAGE_RUN_CODE $MAGE_RUN_CODE;
    fastcgi_param MAGE_RUN_TYPE $MAGE_RUN_TYPE;
    fastcgi_param SCRIPT_FILENAME $MAGE_ROOT/pub$fastcgi_script_name;
    include fastcgi_params;
}

gzip on;
gzip_disable "msie6";

gzip_comp_level 6;
gzip_min_length 1100;
gzip_buffers 16 8k;
gzip_proxied any;
gzip_types
    text/plain
    text/css
    text/js
    text/xml
    text/javascript
    application/javascript

```

```

        application/x-javascript
        application/json
        application/xml
        application/xml+rss
        image/svg+xml;
gzip_vary on;

# Banned locations (only reached if the earlier PHP entry point regexes don't
match)
location ~* (\.php$|\.htaccess$|\.git) {
    deny all;
}
}

```

/etc/nginx/sites_enabled/ext-magento.conf

```

server {
    listen 80 ;
    server_name magento.ulla-yanez.local;
#    return 301 https://$host$request_uri;
    rewrite ^/(.*)$ https://magento.ulla-yanez.local/$1 permanent;
}
server {
    listen 443 ssl;
    server_name magento.ulla-yanez.local;

    access_log          /media/ecommerce/logs/logs_nginx/ext-magento.ulla-yanez.local-
access.log;
    error_log /media/ecommerce/logs/logs_nginx/ext-magento.ulla-yanez.local-error.log;

    send_timeout        600;

    ssl on;
    include snippets/snakeoil.conf;

    location /robots.txt { alias /home/www/html/robots.txt; }

    location ~ /(\.git|\.htaccess|\.ssh) {
        deny all;
        return 403;
    }

    location /{
        include proxy_public.conf;
        fastcgi_buffers 16 16k;
        fastcgi_buffer_size 32k;
        proxy_buffer_size 128k;
        proxy_buffers 4 256k;
        proxy_busy_buffers_size 256k;
        proxy_pass http://127.0.0.1:6081; #A Varnish
        #proxy_pass http://127.0.0.1:8080; #Saltando Varnish por troubleshooting
        add_header Host $host;
    }

    location ~ /admin_1qw7wb {
        include proxy_admin.conf;
        fastcgi_buffers 16 16k;
        fastcgi_buffer_size 32k;
        proxy_buffer_size 128k;
        proxy_buffers 4 256k;
        proxy_busy_buffers_size 256k;
        proxy_pass http://127.0.0.1:8080; #Siempre saltando Varnish
    }
}
}

```

9.10. Archivos de configuración modificados para Magento

/media/ecommerce/current/app/etc/env.php

```

<?php
return [

```

```

'backend' => [
  'frontName' => 'admin_1qw7wb'
],
'remote_storage' => [
  'driver' => 'file'
],
'queue' => [
  'consumers_wait_for_messages' => 1
],
'crypt' => [
  'key' => 'b155b94b190f1a5560bc5ed2d720e3d4'
],
'db' => [
  'table_prefix' => '',
  'connection' => [
    'default' => [
      'host' => '10.20.22.40',
      'dbname' => 'ecommerce',
      'username' => 'ecomm_user',
      'password' => 'VJXsRmwRDGeHsT3m',
      'model' => 'mysql4',
      'engine' => 'innodb',
      'initStatements' => 'SET NAMES utf8;',
      'active' => '1',
      'driver_options' => [
        1014 => false
      ]
    ]
  ]
],
'resource' => [
  'default_setup' => [
    'connection' => 'default'
  ]
],
'x-frame-options' => 'SAMEORIGIN',
'MAGE_MODE' => 'production',
'session' => [
  'save' => 'redis',
  'redis' => [
    'host' => '10.20.22.50',
    'port' => '6379',
    'password' => '',
    'timeout' => '2.5',
    'persistent_identifier' => '',
    'database' => '2',
    'compression_threshold' => '2048',
    'compression_library' => 'gzip',
    'log_level' => '4',
    'max_concurrency' => '6',
    'break_after_frontend' => '5',
    'break_after_adminhtml' => '30',
    'first_lifetime' => '600',
    'bot_first_lifetime' => '60',
    'bot_lifetime' => '7200',
    'disable_locking' => '0',
    'min_lifetime' => '60',
    'max_lifetime' => '2592000',
    'sentinel_master' => '',
    'sentinel_servers' => '',
    'sentinel_connect_retries' => '5',
    'sentinel_verify_master' => '0'
  ]
],
'cache' => [
  'frontend' => [
    'default' => [
      'id_prefix' => '501_',
      'backend' => 'Magento\\Framework\\Cache\\Backend\\Redis',
      'backend_options' => [
        'server' => '10.20.22.50',
        'database' => '0',
        'port' => '6379',
        'password' => '',
        'compress_data' => '1',
      ]
    ]
  ]
]

```

```

        'compression_lib' => ''
    ]
],
'page_cache' => [
    'id_prefix' => '501_',
    'backend' => 'Magento\\Framework\\Cache\\Backend\\Redis',
    'backend_options' => [
        'server' => '10.20.22.50',
        'database' => '1',
        'port' => '6379',
        'password' => '',
        'compress_data' => '0',
        'compression_lib' => ''
    ]
]
],
'allow_parallel_generation' => true
],
'lock' => [
    'provider' => 'db'
],
'directories' => [
    'document_root_is_pub' => true
],
'cache_types' => [
    'config' => 1,
    'layout' => 1,
    'block_html' => 1,
    'collections' => 1,
    'reflection' => 1,
    'db_ddl' => 1,
    'compiled_config' => 1,
    'eav' => 1,
    'customer_notification' => 1,
    'config_integration' => 1,
    'config_integration_api' => 1,
    'full_page' => 1,
    'config_webservice' => 1,
    'translate' => 1
],
'downloadable_domains' => [
    'magento.ulla-yanez.local'
],
'install' => [
    'date' => 'Sun, 04 Dec 2022 10:14:19 +0000'
],
'http_cache_hosts' => [
    [
        'host' => '10.20.22.10',
        'port' => '6081'
    ]
]
];

```