



Sistemas distribuidos. Replicación de datos

Alejandro Montilla Delgado

Grado Ingeniería Informática – Tecnologías de la información

Manuel Jesús Mendoza Flores

8 de junio de 2016

© (el autor/a)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	Sistemas distribuidos. Replicación de datos
Nombre del autor:	Alejandro Montilla Delgado
Nombre del consultor:	Manuel Jesús Mendoza Flores
Fecha de entrega (mm/aaaa):	06/2016
Área del Trabajo Final:	Administración de redes y SO
Titulación:	<i>Grado Ingeniería Informática</i>
Resumen del Trabajo (máximo 250 palabras):	
<p>Se realizará un estudio de sistemas distribuidos de ficheros, basándolo en los sistemas de réplicas de datos distribuidos geográficamente.</p> <p>Mediante este proyecto, se busca, que con la utilización sistemas de archivos distribuidos y réplicas de datos, conseguir dar una contingencia a la información de una empresa que se encuentra distribuida en diversos puntos geográficos, logrando un acceso local a los datos o bien un sistema de protección frente al borrado de los mismos.</p> <p>El proyecto se basará el análisis y configuración de tres soluciones diferentes existentes en el mercado, como primera solución, se analizará el DFS de Microsoft, posteriormente SnapMirror de NetApp y una como tercera solución, Linux con software OpenSource</p>	

Abstract (in English, 250 words or less):

I will be made a distributed file system study, basing this study on systems replications of data distributed geographically.

Through this project, I seeks, that we can use distributed file systems and data replication get an information contingency from a company that is on various geographical locations, managing a data local access or a system protection against deleting them.

The project will be bases on an analysis and configuration of three different solutions existing on the market, the first solution based on Microsoft DFS, NetApp SnapMirror later and the third solution I will analyze OpenSource Linux software

Palabras clave (entre 4 y 8):

Ficheros distribuidos
Contingencia
Replicación
Disaster Recovery

Índice

1	Introducción.....	1
1.1	Contexto y justificación del Trabajo.....	1
1.2	Objetivos del Trabajo	3
1.3	Enfoque y método seguido.....	3
1.4	Planificación del Trabajo	5
1.5	Breve resumen de productos obtenidos.....	8
1.6	Breve descripción de los otros capítulos de la memoria	8
2	Sistemas de archivos distribuidos	9
2.1	Definición de los sistemas de ficheros distribuidos.	9
2.2	Propiedades de los sistemas de ficheros distribuidos	10
3	Estudio de los productos de DFS.....	12
3.1	Distribution File System de Microsoft	13
3.2	SnapMirror de NetApp.....	17
3.3	Solución OpenSource en Linux (DRBD)	20
4	Implementación de los sistemas	22
4.1	Definición del laboratorio.....	22
4.2	Configuración básica del entorno de laboratorio	23
4.3	Instalación y configuración de DFSR	27
4.4	Instalación y configuración de SnapMirror	30
4.5	Instalación y configuración de OpenSource de Linux.....	34
5	Conclusiones.....	37
6	Glosario.....	40
7	Bibliografía	41

Lista de figuras

Figura 3-1: Configuración sin espacio de nombres	13
Figura 3-2: Configuración con espacio de nombres	13
Figura 3-3: Ejemplos de topología Concentrador y radio	15
Figura 3-4: Ejemplos de topología Malla completa.....	16
Tabla 3-1: Comparación SnapMirror por volumen o por Qtree.....	19
Figura 3-5: Ejemplos de topología DRBD	21
Figura 4-1: Esquema de configuración del laboratorio	23
Tabla 4-1: Configuración básica WANem	23
Figura 4-2: Configuración red WANem	24
Figura 4-3: Configuración línea datos mediante WANem	25
Tabla 4-2: Configuración SRV00-DA.....	25
Figura 4-4: Configuración DNS	26
Tabla 4-3: Configuración SRV-11.....	27
Tabla 4-4: Configuración SRV-21.....	27
Figura 4-5: Consola de administración.....	28
Figura 4-6: Selección tipo de grupo de replicación.....	28
Figura 4-7: Selección tipo de grupo de replicación.....	29
Figura 4-8: Prueba básica de replicación	29
Tabla 4-5: Configuración NODO11	30
Tabla 4-6: Configuración NODO21	30
Figura 4-9: Licencia SnapMirror	30
Figura 4-10: Habilitar SnapMirror en el filer.....	30
Figura 4-11: Permisos de acceso al filer origen	31
Figura 4-12: Creación de volumen en destino.....	31
Figura 4-13: Creación de la relación de SnapMirror entre volúmenes	31
Figura 4-14: Configuración en modo síncrono	32
Figura 4-15: Configuración en modo programado.....	32
Figura 4-16: Verificación replica volumen (copia de Qtree).....	32
Figura 4-17: Verificación replica volumen (copia de LUN)	33
Figura 4-18: Creación de la relación de SnapMirror de Qtree	33
Tabla 4-7: Configuración LINUX11.....	34
Tabla 4-8: Configuración LINUX21.....	34
Tabla 4-9: Módulos DRBD.....	34
Figura 4-19: Fichero /etc/drbd.conf	35
Figura 4-20: Fichero /etc/drbd.d/global_common.conf	35
Figura 4-21: Fichero /etc/drbd.d/r0.res	36
Figura 4-22: Inicialización de meta-datos	36
Figura 5-1: Ejemplo convivencia sistemas de réplica.....	37
Figura 5-2: Ejemplo interacción sistemas de réplica	38

1 Introducción

1.1 Contexto y justificación del Trabajo

El proyecto se basará en el análisis y configuración de tres soluciones diferentes existentes en el mercado, como primera solución, se analizará el DFS de Microsoft, posteriormente SnapMirror de NetApp y una como tercera solución, Linux con software OpenSource.

En la actualidad, uno de los problemas que se presentan es la distribución geográfica de los datos. En estos días, la movilidad de los usuarios es cada vez mayor y las empresas tienden a "dispersarse" mediante pequeñas oficinas, bien sean en oficinas fijas o mediante oficinas de obra, lo que genera una gran cantidad de puntos de generación de datos.

Todos estos datos han de protegerse, y la implementación de sistemas de backup fiables no suele ser sencilla dado que por norma general los sistemas completos de backup son complejos, además de necesitar personal preparado para su configuración, puesta en marcha y mantenimiento.

Una de las soluciones es recopilar los datos y centralizarlos en un punto, donde se protegerán mediante los diversos sistemas de backup. La recopilación de datos, se puede realizar mediante sistemas de replicación, desde sistemas sencillos de copia programada, hasta sistemas más complejos de réplica en tiempo real.

Actualmente existen multitud de herramientas que permiten el uso de sistemas distribuidos de ficheros o bien, el realizar réplicas sobre un equipo raíz para la consolidación de toda la información en un mismo punto y de esta forma, facilitar la protección de los datos.

Dentro de la multitud de posibilidades existentes en el mercado, y basando en un caso real vivido, se han seleccionado dos herramientas principales en las que centrar el estudio:

- DFS-R de Microsoft.
Es el sistema de replicación de datos desarrollado por Microsoft a partir de su sistema operativo Windows Server 2003.
Este sistema forma parte del desarrollo de Microsoft DFS que es el sistema distribuido de información (espacio de nombres), podríamos decir que es la funcionalidad del sistema DFS para compartir y replicar información.

- SnapMirror de NetApp.
SnapMirror es el sistema de replicación de datos desarrollado por NetApp, el cual permite la réplica de datos tanto de forma síncrona como asíncrona, pudiendo realizar recuperación de desastres de datos, de una manera eficaz y rápida.

Como solución alternativa a software de pago y pudiendo este ser implementado en pequeñas y medianas delegaciones, se realizará un estudio adicional con una solución Linux con software de replicación OpenSource.

Este software que se analizará en este proyecto será una solución SUSE unido a DRBD como software de replicación, dado que este software actualmente ya se encuentra integrado dentro de esta distribución de Linux.

La implantación de cualquiera de los softwares nombrados anteriormente, permite la convivencia de los tres en un mismo conjunto de servidores de una red de datos.

1.2 Objetivos del Trabajo

Los principales objetivos de este proyecto son:

- El objetivo es conseguir agrupar una copia de toda la información de las sucursales en la sede central, mediante replicación de datos y así poder agregarlas a las políticas de backup.
- Dentro de este objetivo, la solución tiene que estar basada en uno de los sistemas de los que ya dispone la empresa (Microsoft, NetApp o Linux).

Los objetivos parciales para lograrlo son los siguientes:

- Estudio de las características de la solución Microsoft.
- Estudio de las características de la solución de NetApp.
- Estudio de las características NAS de Linux.
- Análisis de convivencia de los tres productos.

Dentro de la diversidad de soluciones que podemos encontrar en el mercado, realizaremos el análisis e implementación de tres soluciones, por un lado, una solución Microsoft; por otro lado, una solución mediante equipos NetApp y como última solución, una basada en Linux con un software NAS OpenSource.

Dentro del estudio, deberemos de tener en cuenta dos factores claves a la hora de tomar una decisión:

- La cantidad máxima de datos que podemos tolerar perder - RPO (Recovery Point Objective)
- Tiempo de recuperación que podemos tolerar - RTO (Recovery Time Objective)

1.3 Enfoque y método seguido

Tras realizar un análisis de la situación y un estudio de las necesidades, para la replicación de datos, lo cual es objeto de este proyecto, me planteé dos diferentes estrategias a seguir:

- Partir de 0, realizando un estudio de un desarrollo completo de un sistema de replicación basado en herramientas existentes.
- Aplicar la experiencia personal en herramientas de replicación, con las cuales he trabajado en circunstancias muy parecidas.

Después de analizar las estrategias anteriores, decido elegir la segunda opción. Esta decisión está motivada la necesidad de profundizar en el conocimiento de las herramientas de replicación, basándome en dos de las más importantes empresas desarrolladoras de sistemas operativos que incluyen estas funcionalidades, como son Microsoft y NetApp.

Adicionalmente, me parece interesante analizar y estudiar una herramienta de software libre como alternativa a problemas presupuestarios que puedan existir a la hora de implantar un sistema de estas características.

Considero que la estrategia adoptada es la más apropiada para conseguir estos objetivos por los siguientes motivos:

- Robustez de las herramientas
- Fiabilidad de los sistemas
- Mantenimiento de las herramientas por parte de las empresas propietarias.
- Evolución de las herramientas por parte de las empresas propietarias.
- Experiencia personal en estas herramientas.
- La herramienta Linux por coste/rendimiento.

1.4 Planificación del Trabajo

Para llevar a cabo el estudio y la creación de este trabajo, se necesitarán las siguientes herramientas:

Equipos informáticos

Debido a la falta de disponibilidad de equipos físicos, el estudio se realizará en un entorno virtualizado mediante VMWare.

Sistemas y Herramientas

- Software de virtualización
- Windows Server 2008/2012
- Maqueta de Directorio Activo
- Servidores de ficheros Windows (Windows 2008/2012)
- Simulador equipos NetApp
- Emulador de tráfico WAN (WANEM)
- Linux
- NAS OpenSource
- Documentación de Comandos de NetApp
- Documentación TechNet de Microsoft
- Documentación Linux

Tarea 1: Análisis y estudio solución Microsoft.

Descripción de la tarea

Estudio completo del entorno DFS de Microsoft

Objetivos de la tarea

Conocer las posibilidades que nos ofrece Microsoft en el sistema de replicación de datos.

Tarea 2: Análisis y estudio solución NetApp.

Descripción de la tarea

Estudio completo del entorno SnapMirror de NetApp

Objetivos de la tarea

Conocer las posibilidades que nos ofrece NetApp en el sistema de replicación de datos.

Tarea 3: Análisis y estudio solución Linux.

Descripción de la tarea

Estudio completo de un entorno OpenSource de Linux

Objetivos de la tarea

Conocer las posibilidades que nos ofrece Linux en el sistema de replicación de datos.

Tarea 4: Diseño de laboratorio.

Descripción de la tarea

Diseño y creación de laboratorio

Objetivos de la tarea

Disponer de un laboratorio de pruebas donde poder realizar pruebas de todo el software a analizar.

Tarea 5: Instalación y configuración de las herramientas a analizar.

Descripción de la tarea

Instalación y configuración de las tres herramientas de replicación seleccionadas.

Objetivos de la tarea

Conocer las posibilidades de implementación de las herramientas y poder disponer de un entorno de pruebas de dichas herramientas.

Tarea 6: Batería de pruebas.

Descripción de la tarea

Realización de todas las pruebas de función del software

Objetivos de la tarea

Conocer el comportamiento de cada uno de los programas que estamos analizando para obtener información de los mismos.

Task Name	Duración	Trabajo	Comienzo	Fin
1 Proyecto	106,13 días	841 horas	mié 24/02/16	mié 08/06/16
2 Recopilación de información	7,88 días	63 horas	mié 24/02/16	mié 02/03/16
3 Recopilación de información + Entrega PEC1	9,88 días	79 horas	mié 02/03/16	vie 11/03/16
4 Comparativa entre tecnologías a usar y diseño teórico del laboratorio	6,88 días	55 horas	sáb 12/03/16	vie 18/03/16
5 Preparación del laboratorio de pruebas	6,88 días	55 horas	sáb 19/03/16	vie 25/03/16
6 Instalación del software necesario	6,88 días	55 horas	sáb 26/03/16	vie 01/04/16
7 Configuración de los sistemas Microsoft y pruebas	6,88 días	55 horas	sáb 02/04/16	vie 08/04/16
8 Entrega PEC2: herramientas elegidas + diseño base	6,88 días	55 horas	sáb 09/04/16	vie 15/04/16
9 Instalación de los simuladores NetApp	6,88 días	55 horas	sáb 16/04/16	vie 22/04/16
10 Configuración de NetApp y pruebas	6,88 días	55 horas	sáb 23/04/16	vie 29/04/16
11 Instalación de Linux y software OpenSource	6,88 días	55 horas	sáb 30/04/16	vie 06/05/16
12 Configuración de solución Linux y pruebas	6,88 días	55 horas	sáb 07/05/16	vie 13/05/16
13 Entrega PEC3: resultados	6,88 días	55 horas	sáb 14/05/16	vie 20/05/16
14 Cumplimentación memoria	6,88 días	55 horas	sáb 21/05/16	vie 27/05/16
15 Cumplimentación memoria	10,88 días	87 horas	sáb 28/05/16	mar 07/06/16
16 Entrega TFG y Presentación	0,88 días	7 horas	mié 08/06/16	mié 08/06/16

1.5 Breve resumen de productos obtenidos

Con la realización de este proyecto, no se ha obtenido un producto como tal. El resultado ha sido tres sistemas de réplica de datos preparados para una contingencia en caso de desastre de pérdida de datos.

1.6 Breve descripción de los otros capítulos de la memoria

1. Introducción: Resumen, planificación y procedimientos a seguir a lo largo del estudio del proyecto.
2. Sistemas de archivos distribuidos: Definición, propiedades y conceptos de los sistemas de archivos distribuidos genéricos.
3. Estudio de los productos de réplica: Estudio específico de los tres sistemas seleccionados para la realización del proyecto.
4. Implementación de los sistemas: Puesta en marcha, estudio y pruebas de los sistemas distribuidos listados en el capítulo 3.

2 Sistemas de archivos distribuidos

2.1 Definición de los sistemas de ficheros distribuidos.

El sistema de archivos distribuido (Distributed File System - DFS) es un sistema que permite el acceso a los archivos y carpetas desde numerosos puntos. Este sistema es una mejora del modelo clásico de compartición de ficheros y recursos de almacenamiento entre múltiples usuarios en una misma red. Este sistema de archivos distribuido realiza esta misma base de comportamiento, pero además permite realizarla cuando los usuarios están dispersos geográficamente en diversos sitios.

El origen de los sistemas de archivos distribuidos se remonta a los años 70, pero no es hasta 1984 cuando aparece el tan conocido y actualmente utilizado sistema NTF (Network File System), creado por Sun Microsystems para SunOS, siendo este el primer sistema comercial de archivos en red.

Los sistemas de ficheros distribuidos proporcionan servicios de archivo remoto a clientes, de forma transparente, estando los clientes, servidores, y dispositivos de almacenamiento entre las máquinas de un sistema distribuido, permitiendo la compartición de datos entre los usuarios.

Además de la compartición de datos, un sistema de archivos distribuido busca una serie de objetivos, como es la disponibilidad, el rendimiento, la transparencia o la tolerancia a fallos.

Uno de los aspectos más importantes en un sistema de archivos distribuidos, es el cómo tratar el que existan varios clientes trabajando sobre un mismo fichero. En este caso, se pueden encontrar tres formas diferentes a abordar este problema:

- **Control de Sesión:** En este caso, los cambios de un fichero abierto son únicamente visible para quien lo modifico y no es hasta que se cierra el archivo, cuando estos cambios son actualizados en el servidor. En el caso de que existan dos sesiones concurrentes sobre un mismo fichero, será la última en cerrar la que actualice los cambios.
- **Fichero de solo lectura:** En este caso, todo el contenido existente está en solo lectura, por lo que en ningún caso habrá modificaciones.
- **Transacciones:** El sistema ejecuta las transacciones de los ficheros de forma secuencial según el orden de llegada, realizando todo en control de transacciones mediante "BEGIN TRANSACTION" y "END TRANSACTION"

Otro de los aspectos fundamentales en un sistema de ficheros es el método de acceso remoto, es decir, como se realiza el acceso a los ficheros que no están en nuestro sistema, para poner en servicio este acceso, existen diferentes modelos, entre los que destacan los siguientes:

- **Modelo carga/descarga:** En este modelo se realiza una transferencia completa del fichero al cliente antes de abrirlo y se suele utilizar de manera conjunta con la semántica de sesión. Este modelo es muy eficaz en las transferencias ya que se realiza solo una petición remota, y luego los cambios son locales, pero a cambio el coste de abrir un fichero es mayor.
- **Modelo de servicios remotos:** Otro de los modelos es donde se produce un acceso por bloques según se vayan necesitando y las peticiones se almacenan en el servidor, aumentando de ese modo el número de peticiones, pero siendo estas más ligeras, se puede seguir considerando un modelo básico cliente/servidor.
- **Modelo de caché de cliente:** Es una combinación de los dos modelos anteriores.

2.2 Propiedades de los sistemas de ficheros distribuidos

El sistema distribuido de ficheros se implementa mediante un conjunto de máquinas servidores conectadas a través de una red de comunicaciones, permitiendo a diversos clientes acceder a dichos ficheros.

En algunos de los sistemas de ficheros distribuidos, no se necesita el uso de unidades de almacenamiento lógicas, se puede utilizar una organización jerárquica de directorios. Esto es la posibilidad de incorporar en un sistema de ficheros existente un nuevo punto de montaje, el cual se verá como un directorio más de dicha estructura. Por supuesto, todo sistema de archivo, tiene por detrás un sistema de almacenamiento físico.

Los sistemas de ficheros distribuidos tienen las siguientes propiedades genéricas:

- Permite acceso a ficheros de forma transparente para los usuarios.
- No se necesita conocer la localización de los servidores.
- Se puede modificar la ubicación de los datos sin tener que informar a los usuarios de dicho cambio.
- Permite replicación de archivos a varios puntos.
- Admite tolerancia a fallos

En el mercado, actualmente existen numerosas aplicaciones para el uso de archivos distribuidos, sobre todo en la parte réplica de datos. Dentro de esta variedad, están desde las aplicaciones más básica, aunque potentes, como puede ser Robocopy a sistemas más complejos como es el caso de Snapmirror, tanto gratuitas como de pago. Algunas de las aplicaciones que se pueden encontrar en el mercado son:

- Robocopy
- Distribution File System Replicated (Microsoft)
- Snapmirror (NetApp)
- DRBD Linux
- Symantec Replicator
- EMC replication
- Comm Vault Simpana Replication
- Zerto Virtual Replication

3 Estudio de los productos de DFS

Existe una diversidad de sistemas de archivos distribuidos, tales como NFS, AFS, CODA, OPEN GFS o GLUSTERFS.

- NFS: El sistema NFS, fue desarrollado por Sun Microsystems en 1984, es un sistema que permite compartir los datos entre varios clientes de forma sencilla y efectiva. Este sistema está creado para redes de área local, posibilitando el acceso a ficheros remotos como si fuera en local. Este sistema tiene una interfaz bien definida entre clientes y servidores, de esta manera puede soportar sistemas heterogéneos, donde clientes y/o servidores ejecutan diferentes sistemas operativos, esto es gracias a la definición de los protocolos cliente-servidor.
- AFS: El sistema de archivos distribuidos Andrew File System, nace como parte del proyecto llamado Proyecto Andrew, actualmente existe varias implementaciones de AFS: Transarc (IBM), OpenAFS, Arla e incluso una implementación realizada por Red-Hat. AFS utiliza kerberos para la autenticación, implementando listas de control de acceso en directorios y ficheros. Cuando un cliente abre un archivo este es cacheado en el equipo cliente, marcando el archivo y en caso de que sea modificado por otro cliente, se avisará a quién lo tenga abierto, realizándose únicamente sobre esta caché las operaciones de lectura y escritura. En el momento de cerrar el fichero, los cambios realizados son copiados al servidor
- CODA: Este sistema de archivos distribuidos fue desarrollado en 1987, siendo descendiente del anteriormente visto sistema AFS. Este proyecto, es de software libre, ofreciendo un gran rendimiento debido a la caché persistente en el cliente, lo que además permite trabajar sin conexión, esto, además facilita el continuar trabajando en caso de fallos de red o caídas intermitentes de la conexión. El sistema dispone de un modelo de seguridad para autenticación y control de acceso, dicho modelo es propio e independiente del sistema operativo. A diferencia de AFS, CODA sí que permite que todos los servidores recibir actualizaciones de los ficheros que se replican.
- GLUSTERFS: Sistema de archivos distribuido, el cual que permite agregar un número determinados de nodos a un espacio para crear un sólo espacio de direccionamiento de ficheros. Actualmente, GLUSTERFS, cuenta además con un sistema de replicación con protección contra fallo de un nodo, es decir, la información no sólo está distribuida a lo largo de diversos nodos, sino que existen varias copias de la misma, de forma que si algún nodo falla, el conjunto sigue funcionando como si nada hubiera pasado, puesto que existe otra copia de la información disponible a la que se puede acceder mientras se recupera el nodo dañado, y una vez recuperado, este se sincroniza y todo el conjunto continúa operando como en un inicio.

Además de los listados anteriormente, existen otros como el DFSR de Microsoft o el SnapMirror de NetApp, los cuales se detallarán más adelante.

3.1 Distribution File System de Microsoft

Dentro de los sistemas de ficheros distribuidos se pueden encontrar dos tipos diferenciados, el “espacio de nombres de DFS” y la replicación de datos (DFSR), aunque DFS necesita de la replicación de datos para su correcto funcionamiento.

El “espacio de nombres DFS”, posibilita agrupar las carpetas compartidas en diferentes servidores. De esta manera, los usuarios ven cada espacio de nombres como una sola carpeta compartida con una serie de subcarpetas.

En un sistema donde no exista una configuración de “Espacio de nombres”, los clientes deberán de conocer cada uno de los servidores donde se encuentran alojados los datos a los que se desea acceder.

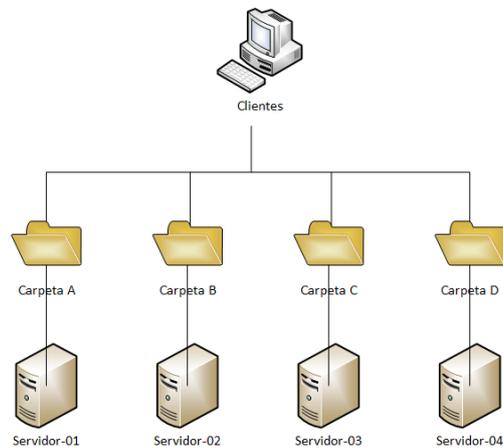


Figura 3-1: Configuración sin espacio de nombres

En el caso de que exista un espacio de nombres, el cliente únicamente deberá de conocer el nombre del espacio de nombres, dentro de esta carpeta, los usuarios verán todas las carpetas compartidas existentes en esta estructura lógica. Proporciona transparencia en la ubicación.

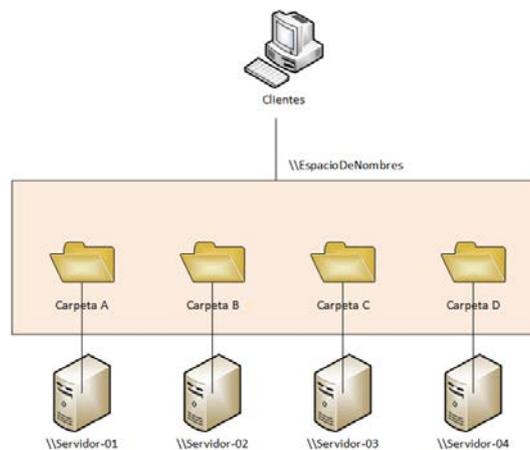


Figura 3-2: Configuración con espacio de nombres

Existen dos tipos de sistemas de espacios de nombres, el independiente y el de dominio. El espacio de nombres independiente, almacena toda la información de la topología en el registro local, por lo que, en caso de caída de esta máquina, todo el sistema cae.

En el caso del DFS de domino, la topología está almacenada en el directorio activo, lo que proporciona una tolerancia a fallo, al estar las carpetas compartidas en diversos servidores. Este tipo, además admite la posibilidad de réplica de datos, por lo que el sistema gana un nivel más de seguridad en tolerancia a fallos.

La replicación es otra de las funciones que posee el DFS de Microsoft y que como ya se ha comentado, el espacio de nombre necesita el espacio de nombres para funcionar correctamente.

La replicación garantiza la disponibilidad de las carpetas DFS en otro servidor en caso de caída del servidor principal.

A la hora de configurar la replicación, entra en juego un concepto muy importante, RPO (Recovery Point Objective). El RPO es el volumen de datos que una empresa se puede permitir perder. En función de este valor, se deberá de configurar los tiempos de réplicas, es decir, si la empresa no se puede permitir la pérdida de casi ningún dato, la réplica se deberá de configurar en tiempo real, esto penalizaría la línea de comunicaciones e incluso la red del propio servidor, pero frente a un fallo del servidor existirá una copia casi actual de los datos en otro servidor.

En el supuesto de que la empresa valore que prefiere penalizar una supuesta pérdida de datos frente a una saturación de la red, la réplica se puede configurar a unas horas determinadas y no afectando en los horarios de trabajo. Todo dependerá de la naturaleza del negocio.

La replicación DFS utiliza un algoritmo de compresión llamado Compresión Diferencial Remota (RDC). Este algoritmo es capaz de detectar los cambios en un determinado fichero y replicar únicamente los bloques modificados del archivo.

Existen dos tipos de grupos de replicación:

- Grupo de replicación para recopilación de datos: Tipo de grupo de replicación para que solo configurable entre dos nodos. Este tipo solo permite la conexión de dos direcciones, pero podría configurarse de forma única.
 - Conexión de una dirección: La información insertada en el Nodo A, se replica sobre el Nodo B o, al contrario, pero solo en una dirección.
 - Conexión de dos direcciones: La información insertada en el Nodo A se replica en el Nodo B y la información insertada en el Nodo B se replica sobre el Nodo A.
- Grupo de replicación multipropósito: Este grupo permite más de dos Nodos. la replicación se realiza grupos de servidores (una pareja de servidores también se considera grupo).

Para la replicación DFS multipropósito se necesita configurar la topología, que son las conexiones lógicas existentes a utilizar. Existen tres tipos diferentes de topologías:

- Concentrador y radio: En esta topología, se puede disponer de hasta dos concentradores, el resto de nodos serán radios. La replicación se realizará desde los nodos radiales a los concentradores o viceversa, pero nunca entre radiales. Para este tipo de topología, se necesitan configurar al menos tres nodos.

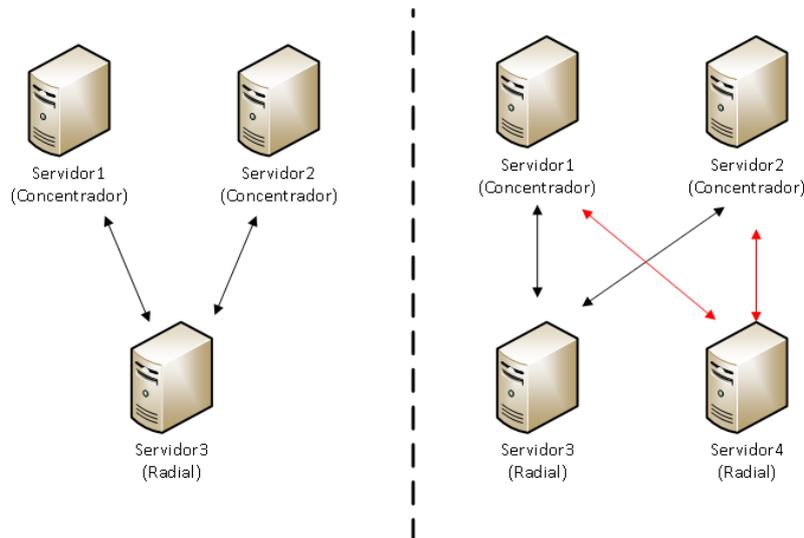


Figura 3-3: Ejemplos de topología Concentrador y radio

- Malla completa: En este tipo de topología, todos los nodos replican contra todos. Existe una recomendación para el correcto funcionamiento de un límite de 10 nodos.

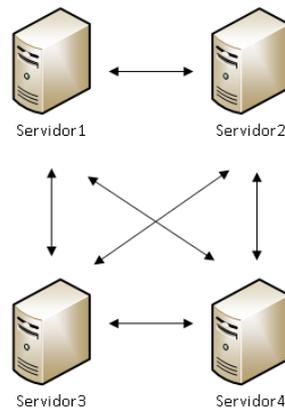


Figura 3-4: Ejemplos de topología Malla completa

- Sin topología: Configuración de topología manual. El asistente de creación del grupo de réplica, se realiza sin configurar ninguna topología. Una vez finalizada la creación del grupo, la réplica no comenzará hasta que no se configure la topología. Esta se deberá de configurar a mano.

3.2 SnapMirror de NetApp

El sistema SnapMirror de NetApp es el sistema por el cual, se puede realizar replicación de datos de forma síncrona y asíncrona, además de recuperación de desastres de datos, de forma eficiente y rápida. Debido al amplio número de años que lleva en el mercado, ha conseguido evolucionar hasta utilizar el ancho de banda de una forma muy eficiente.

La principal función para la que fue creada SnapMirror es realizar una réplica de datos entre localizaciones para protección de los mismos frente a desastres.

Se puede considerar que SnapMirror es eficiente, dado que la replicación por bloques (una de las opciones que tiene), reduce el tráfico de red así como el tiempo de replicación necesario.

Su configuración de réplica es flexible. SnapMirror puede configurarse en la opción One-to-One, One-To-Many, Many-To-One y Many-To-Many.

SnapMirror tiene dos modos operativos de funcionamiento, uno es mediante la replicación de volúmenes y el segundo es mediante la replicación de Qtrees. Ambos métodos, tienen como resultado una copia exacta en destino, la cual se puede habilitar en modo escritura, algo muy útil en caso de fallo en el origen, ya que de una forma rápida se podrá habilitar un repositorio nuevo hasta poder subsanar el fallo en origen. Una vez solucionado el problema en el origen, se realizaría una sincronización de recuperación tras fallo para copiar los cambios y así poder restaurar el funcionamiento normal.

- Replicación por volúmenes: La replicación de volúmenes se realiza a nivel físico de bloque. Esto replica el contenido completo de un volumen, incluyendo LUNs, Qtrees, Shares e incluso si en el volumen original está habilitado el sistema de deduplicación, este sería heredado por el volumen de destino. Condición indispensable para la utilización de réplica por volumen, es que la versión de Data OnTap en el equipo de destino sea igual o superior que la del equipo de origen.
- Replicación para Qtree: En este caso SnapMirror replica de forma individual los Qtree seleccionados. Dado que los Qtrees son subgrupos dentro de un volumen, la operación de réplica, en este caso será a nivel lógico. Es implica que no se puede únicamente replicar un Qtree como tal, dado que faltaría información de contabilidad a nivel del documento Qtree en el sistema de destino.

La replicación a nivel lógico, implica varias diferencias de la réplica de Qtrees respecto a la réplica por volúmenes:

- SnapMirror para Qtree es menos eficiente a nivel de red respecto a la de volúmenes. La réplica de Qtrees no hereda los ahorros de la deduplicación, es debido a que la deduplicación funciona a nivel de bloque y la réplica se realiza a nivel lógico.
- La réplica de Qtrees no mantiene las funcionalidades de copia de Snapshot. SnapMirror para Qtree, solo replica la última copia de Snapshot existente en el origen, manteniendo en destino solo el par de copias necesarias para realizar las actualizaciones de replicación.

Aunque esto no depende de si la réplica lógica o física, otro aspecto que diferencia ambas réplicas sería el tipo de sincronización (asíncrona, semisíncrona y síncrona). Ambas réplicas realizan una copia inicial en la que se replican todos los datos (volumen o Qtree), y una vez terminada dicha copia, el resto se realiza de forma regular. La réplica de volúmenes se puede realizar asíncrona, semisíncrona o síncrona, mientras que la réplica de Qtrees solo es compatible con la replicación asíncrona.

- Modo asíncrono: SnapMirror replica de forma periódica las copias de los Snapshot del volumen o del Qtree de origen al destino en intervalos de tiempo. En el ciclo de replicación solo se transfieren al destino los bloques nuevos o modificados, lo que da eficiencia a este método en términos de sobrecarga del sistema de almacenamiento y ancho de banda.
- Modo síncrono: Envía actualizaciones desde el origen al destino a medida que se generan cambios en los datos y tan pronto sea posible. Esto ayuda a que los datos escritos en el sistema de origen queden protegidos en el destino incluso si falla el sistema de origen al completo. Para este modo de sincronización, se utiliza el reenvío de NVLOG y del punto de consistencia (CP) para mantener el destino completamente actualizado. El reenvío de NVLOG permite que los datos del registro de escritura que normalmente se almacenan en caché en NVRAM en un almacenamiento de NetApp se sincronicen con el destino. El reenvío del punto de consistencia permite que las imágenes de sistema de archivo en disco se mantengan sincronizadas.
- Modo semisíncrono: Este modo replica al destino los cambios realizados en el origen cada poco tiempo. De este modo, se equilibra los beneficios de una replicación síncrona con los beneficios de rendimiento de una replicación asíncrona. Por un lado, no se requiere espera para que la escritura en el origen se produzca en el destino, y, además, no se utiliza el reenvío de NVLOG. Estos dos cambios aceleran la respuesta de la aplicación con una pequeña coincidencia en cuando al objetivo de punto de recuperación (RPO) que se puede conseguir.

SnapMirror para volúmenes	SnapMirror para Qtree
Física	Lógica
Síncrona o asíncrona.	Asíncrona
Volumen de destino solo de lectura y restringido.	Qtree de destino solo lectura, pero el volumen debe de estar on-line y en modo escritura
Replica todo el volumen, incluyendo Qtrees	Replica solamente los Qtrees configurados
Volumen de destino con misma capacidad que el origen, independientemente del espacio ocupado	Qtree de destino con el espacio suficiente para los datos de origen
El destino hereda los ahorros en deduplicación y los de red	El destino no hereda los ahorros en deduplicación

Tabla 3-1: Comparación SnapMirror por volumen o por Qtree

3.3 Solución OpenSource en Linux (DRBD)

DRBD o Distributed Replicated Block Device es un software de replicación de dispositivos de bloque (discos duros, particiones, volúmenes, etc.) creando un RAID1 a través de la red en distintos servidores en distintas localizaciones.

El sistema de replicación de DRBD es un sistema en tiempo real, realizando una replicación continua, siendo transparente para el uso de los datos que se encuentran en el origen.

DRBD soporta tres modos distintos de replicación síncrono, semisíncrono y asíncrono:

- **Modo Síncrono:** La replicación síncrona establece que hasta que ambos servidores no han confirmado la operación de escritura en el disco tanto local como remoto, no se considera completada la operación de escritura local en el nodo origen. De este modo, ambos nodos tienen siempre la misma información, por lo que en caso de caída de uno de ellos se garantiza la no pérdida de información. Este modo es el recomendado para servidores situados en la misma LAN.
- **Modo semisíncrono:** En este caso, la operación de escritura local se considera completada, cuando el nodo origen confirma la escritura y se envían los datos hacia el nodo destino, sin necesidad de esperar confirmación de escritura por parte de este último. En este caso, en caso de desastre, únicamente se perdería la última información grabada en el nodo origen. En el caso de realizar una parada forzada de un nodo, no existiría pérdida de información, ya que la información ya habría sido enviada.
- **Modo asíncrono:** En este modo, una vez que se ha confirmado la operación de escritura local y el paquete ha sido puesto en la cola de envío, se da por completada la operación. En caso de parada forzada del nodo, toda la información puesta en la pila TCP y que no haya sido enviada, se perdería en el destino. Sin embargo, para replications entre servidores en localizaciones remotas, se recomienda su uso.

DRBD permite la replicación de cualquier dispositivo de bloques, pudiendo incluir en la réplica:

- Disco duro físico
- Partición de disco duro
- Volumen lógico
- Disco USB

En general cualquier dispositivo que sea accesible desde el mapeador de discos de Linux.

Un ejemplo de configuración de DRBD sería el siguiente:

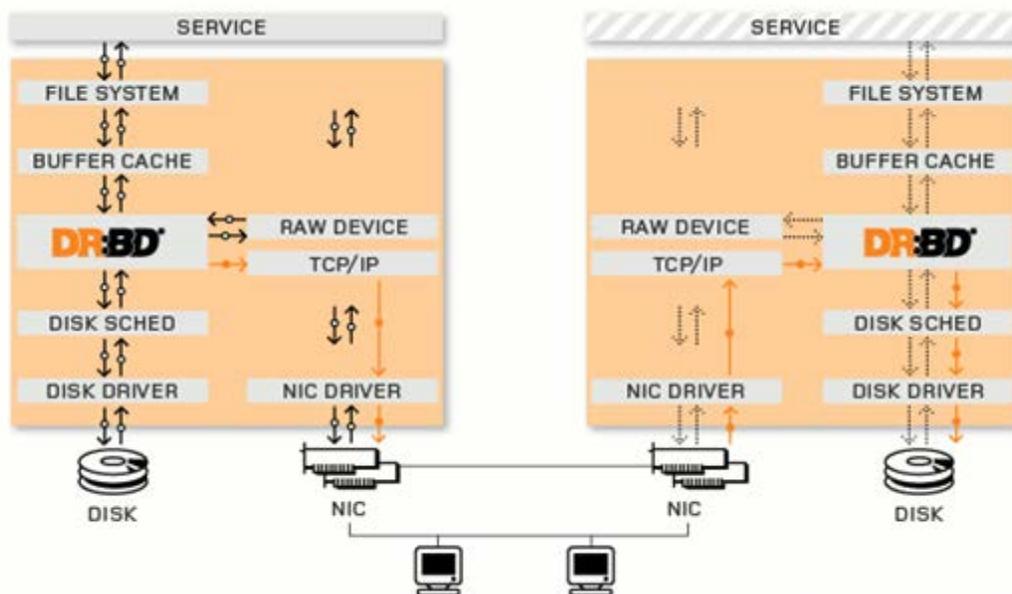


Figura 3-5: Ejemplos de topología DRBD

Imagen propiedad de <http://www.drbd.org/>

4 Implementación de los sistemas

Para poder llevar a cabo las pruebas de funcionamiento de un sistema distribuido de ficheros, se necesitará el diseño e instalación de un entorno de pruebas, intentando ser lo más fiel posible al funcionamiento de una red básica de una empresa de tamaño medio.

4.1 Definición del laboratorio

Para el entorno de laboratorio, se instalará una red WAN compuesta por dos redes LAN, en las cuales, los equipos existentes, pertenecerán todos al mismo dominio (labmon.es – nombre de dominio ficticio para este laboratorio) todo el laboratorio se montará sobre el sistema de virtualización VMWare.

En la primera LAN se instalará un servidor que realizará las funciones de DNS y controlador de domino, un servidor Windows con función de uno de los miembros del sistema DFS, un equipo NetApp y un equipo Linux. Además de la parte de servicios, se necesitará al menos de un equipo cliente para poder realizar pruebas de acceso.

En la segunda red LAN, se instalará un servidor Windows con función del otro de los miembros del sistema DFS, un segundo equipo NetApp y un segundo equipo Linux. Al igual que en la otra LAN, será necesaria la presencia de un equipo cliente.

Dado que se pretende simular una separación geográfica entre las dos redes, se implementará un equipo que sea capaz de emular dicha línea. En este caso, se utilizará WANem.

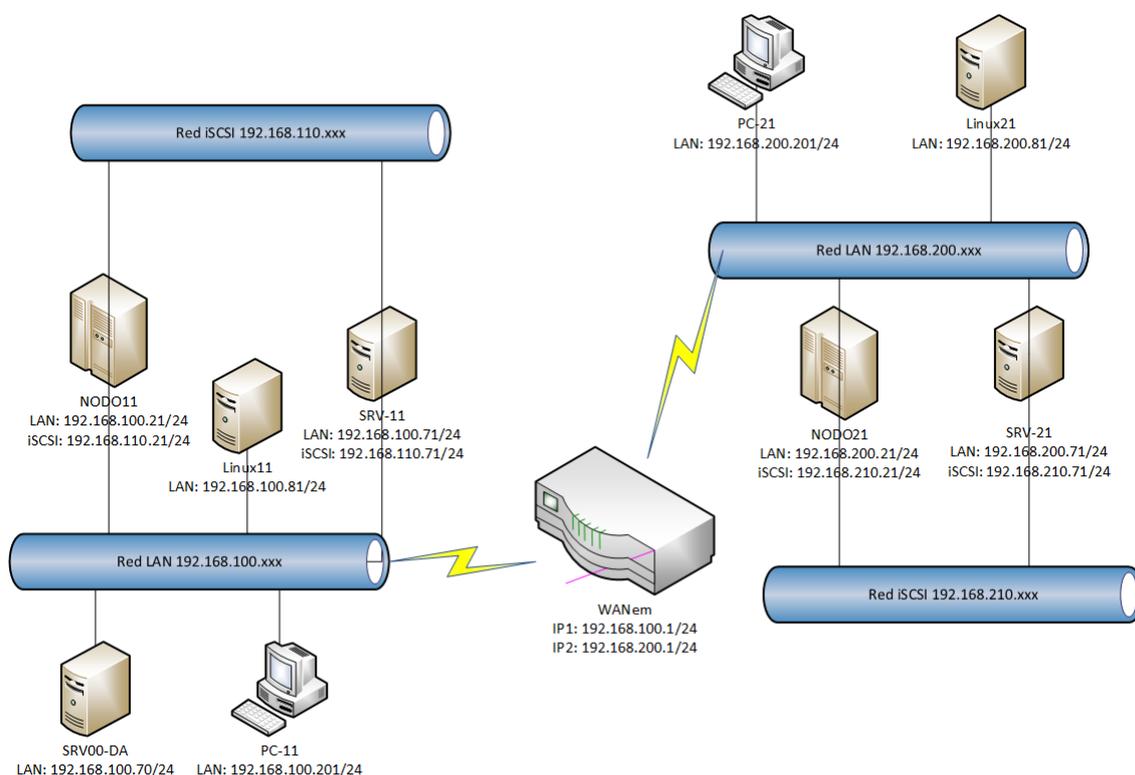


Figura 4-1: Esquema de configuración del laboratorio

4.2 Configuración básica del entorno de laboratorio

4.2.1 Configuración WANem

El primer equipo a configurar será el encargado de emular la línea de comunicaciones. La configuración básica será de un equipo con dos tarjetas de red y el sistema WANem.

WANem	Servicios	WANem
	IP LAN1	192.168.100.1/24
	IP LAN2	192.168.200.1/24

Tabla 4-1: Configuración básica WANem

En este caso se utilizará un equipo con un “sistema cerrado” basado en Unix, disponible de forma gratuita en <http://wanem.sourceforge.net/> una vez descargada la imagen se montará como unidad virtual de CD/DVD en VMWare, se agrega una segunda tarjeta de red (por defecto viene únicamente con una) y se inicia el arranque de la máquina. El equipo viene instalado, únicamente se deberá de configurar las tarjetas de red.

```

IP Settings
=====
eth0      Link encap:Ethernet HWaddr 00:0c:29:a7:a8:60
          inet addr:192.168.100.1 Bcast:192.168.100.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fea7:a860/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:3031 errors:0 dropped:0 overruns:0 frame:0
          TX packets:60 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:370811 (369.9 KiB) TX bytes:11409 (11.1 KiB)
          Interrupt:17 Base address:0x1400

eth1      Link encap:Ethernet HWaddr 00:0c:29:a7:a8:6a
          inet addr:192.168.200.1 Bcast:192.168.200.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fea7:a86a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:2752 errors:0 dropped:0 overruns:0 frame:0
          TX packets:37 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:350420 (342.2 KiB) TX bytes:10226 (9.9 KiB)
          Interrupt:18 Base address:0x1400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:173 errors:0 dropped:0 overruns:0 frame:0
          TX packets:173 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:97622 (95.3 KiB) TX bytes:97622 (95.3 KiB)
=====

```

Figura 4-2: Configuración red WANem

Una vez configuradas las tarjetas, desde un equipo cliente, se accederá a la web <http://192.168.100.1/WANem> desde la cual se podrá realizar la configuración de las líneas. En este caso, la línea será de 10Mbps simétrica con una latencia de 60ms.

WANem commands successfully created									
WANem is running Stop WANem									
Interface: eth1		Packet Limit <input type="text" value="1000"/> (Default=1000)				Symmetrical Network: <input type="text" value="Yes"/>			
Bandwidth		Choose BW <input type="text" value="Thin Ethernet, CAT-3 cable - 10 Mbps"/>				Other: Specify BW(Kbps) <input type="text" value="10240"/>			
Delay		Loss		Duplication		Packet reordering		Corruption	
Delay time (ms)	<input type="text" value="60"/>	Loss(%)	<input type="text" value="0"/>	Duplication (%)	<input type="text" value="0"/>	Reordering (%)	<input type="text" value="0"/>	Corruption(%)	<input type="text" value="0"/>
Jitter(ms)	<input type="text" value="0"/>	Correlation (%)	<input type="text" value="0"/>	Correlation (%)	<input type="text" value="0"/>	Correlation (%)	<input type="text" value="0"/>		
Correlation (%)	<input type="text" value="0"/>					Gap(packet)	<input type="text" value="0"/>		
Distribution	<input type="text" value="-N/A-"/>								
Idle timer Disconnect	Type	<input type="text" value="none"/>	Idle Timer			Disconnect Timer			
Random Disconnect	Type	<input type="text" value="none"/>	MTTF Low	<input type="text"/>	MTTF High	<input type="text"/>	MTR Low	<input type="text"/>	MTR High
Random connection Disconnect	Type	<input type="text" value="none"/>	MTTF Low	<input type="text"/>	MTTF High	<input type="text"/>	MTR Low	<input type="text"/>	MTR High
IP source address	<input type="text" value="any"/>	IP source subnet	<input type="text"/>	IP dest address	<input type="text" value="any"/>	IP dest subnet	<input type="text"/>	Application port if any	<input type="text" value="any"/>

Figura 4-3: Configuración línea datos mediante WANem

4.2.2 Configuración del controlador de dominio

Una vez instalado el sistema operativo (se ha utilizado Windows 2008 R2). Este equipo tendrá dos servicios, controlador de dominio y servidor de DNS.

SRV00-DA	Servicios	Directorio Activo
		DNS
	IP LAN	192.168.100.70/24
	Gateway	192.168.100.1
	HDD Local	Disco C - 20Gb

Tabla 4-2: Configuración SRV00-DA

Lo primero es instalar los roles de DNS y de Active DirectoryDomainServices.

La instalación de ambos roles no se puede hacer en un mismo paso, por lo que se deberá de realizar en dos. Una vez instalados, lo primero en configurar el servicio DNS, requisito indispensable para la instalación del controlador de dominio.

Una vez configurado el DNS, mediante la consola, crearemos la zona de búsqueda directa con nombre dominio labmon.es y posteriormente las zonas de búsqueda inversa 192.168.100.x y 192.168.200.x

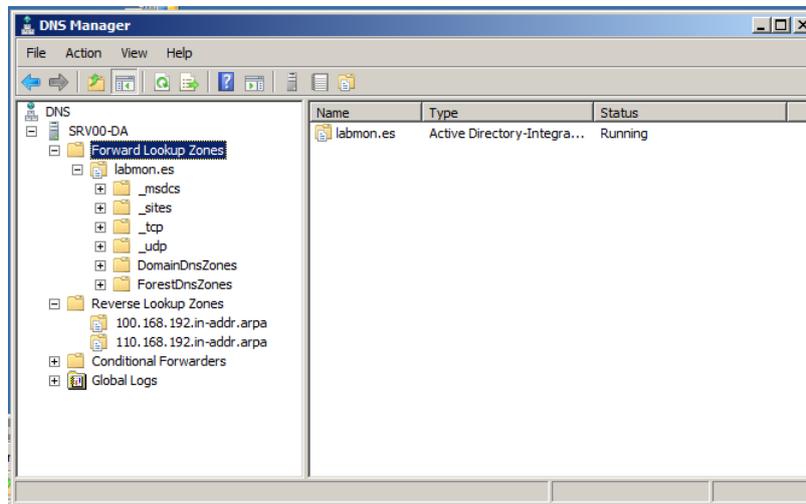


Figura 4-4: Configuración DNS

Por el momento, tanto la zona directa como inversa, serán de tipo primario, una vez que esté el dominio creado, pasarán a estar integradas en el dominio.

En este paso, se deberá de configurar la IP del servidor DNS en la tarjeta de red del servidor y verificar su correcto funcionamiento, como por ejemplo con la herramienta nslookup.

El siguiente paso es crear el crear el dominio y promocionar el servidor a controlador de dominio. Para iniciar la configuración del dominio, se ejecutará el comando DCPROMO en el Símbolo de Sistema. Los datos que se han tenido en cuenta han sido:

- Nuevo bosque.
- Nombre del dominio labmon.es
- Nivel funcional Windows 2008 R2
- No se configura delegación de zona DNS automáticamente
- Integración de zonas DNS en el directorio activo.

4.3 Instalación y configuración de DFSR

Una vez instalado y funcionando el dominio, es el momento de comenzar con la instalación de los equipos encargados del DFSR (SRV-11 y SRV-21). Ambos equipos llevarán una configuración básica idéntica (salvo nombre y direcciones IP).

SRV-11	Servicios	DFS
		File Server
	IP LAN	192.168.100.71/24
	Gateway	192.168.100.1
	IP iSCSI	192.168.110.71/24
	HDD Local	Disco C - 20Gb
	HDD iSCSI	Disco D - 100Mb

Tabla 4-3: Configuración SRV-11

SRV-21	Servicios	DFS
		File Server
	IP LAN	192.168.200.71/24
	Gateway	192.168.200.1
	IP iSCSI	192.168.210.71/24
	HDD Local	Disco C - 20Gb
	HDD iSCSI	Disco D - 100Mb

Tabla 4-4: Configuración SRV-21

El primer paso para será la instalación del Rol “File Services” que engloba “File Server”, “DFS NameSpaces” y “DFS Replication”, aunque se pueden seleccionar individualmente, en este caso serán necesarios todos.

Se comenzará con una configuración únicamente de replicación. Por el momento se omite el espacio de nombre. Una vez instalado el rol, a través de la consola de administración, sobre se indicará un nuevo grupo de replicación.

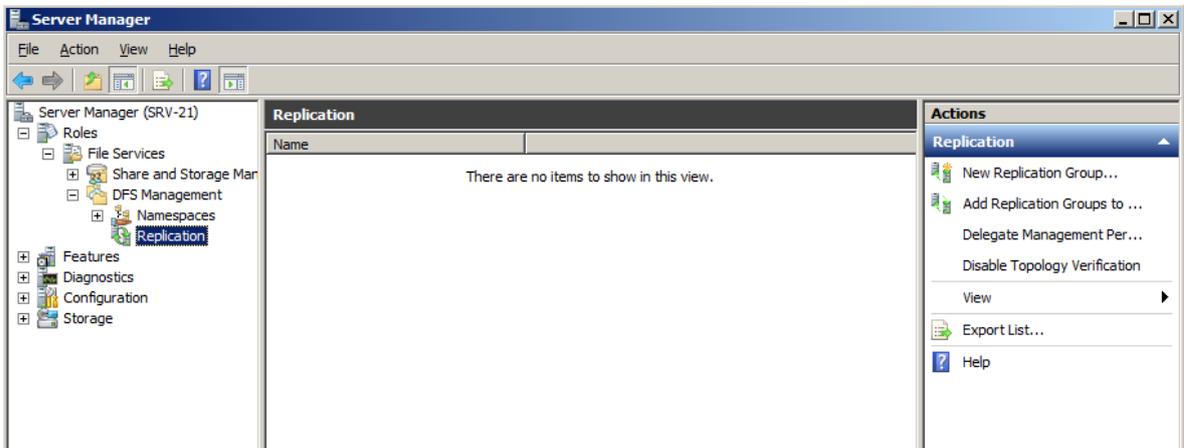


Figura 4-5: Consola de administración

Para esta primera prueba y dado que únicamente están disponibles dos servidores, será un grupo de réplica para recolección de datos.

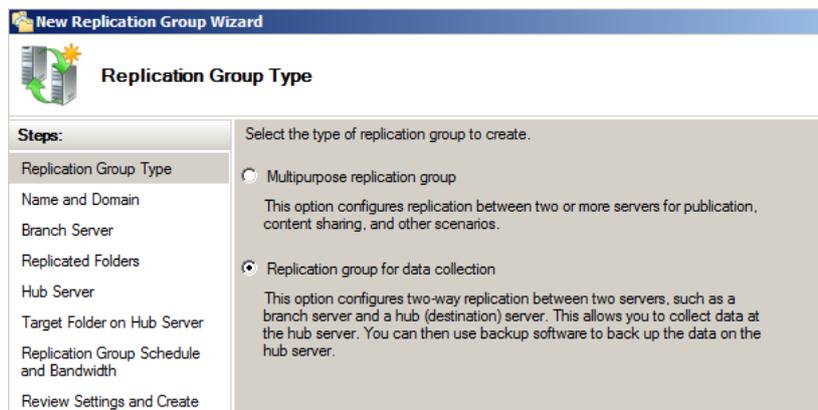


Figura 4-6: Selección tipo de grupo de replicación

Se indicará el servidor origen con la carpeta a replicar, el servidor destino con la carpeta de destino y el ancho de banda y horario de réplica.

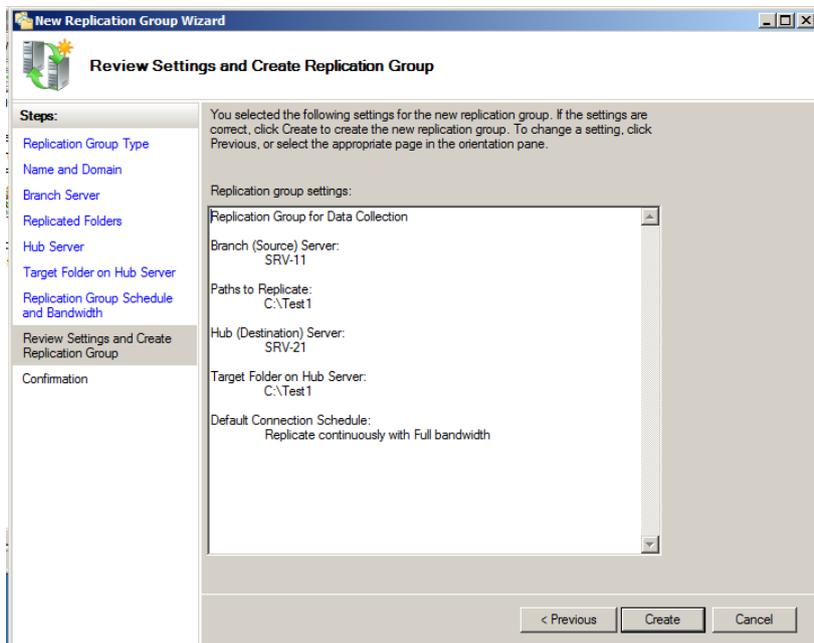


Figura 4-7: Selección tipo de grupo de replicación

Una vez terminado el asistente y creado el grupo, se ha creado un pequeño fichero de texto en SRV-11 y en unos segundos, el fichero ha sido replicado a SRV-21.

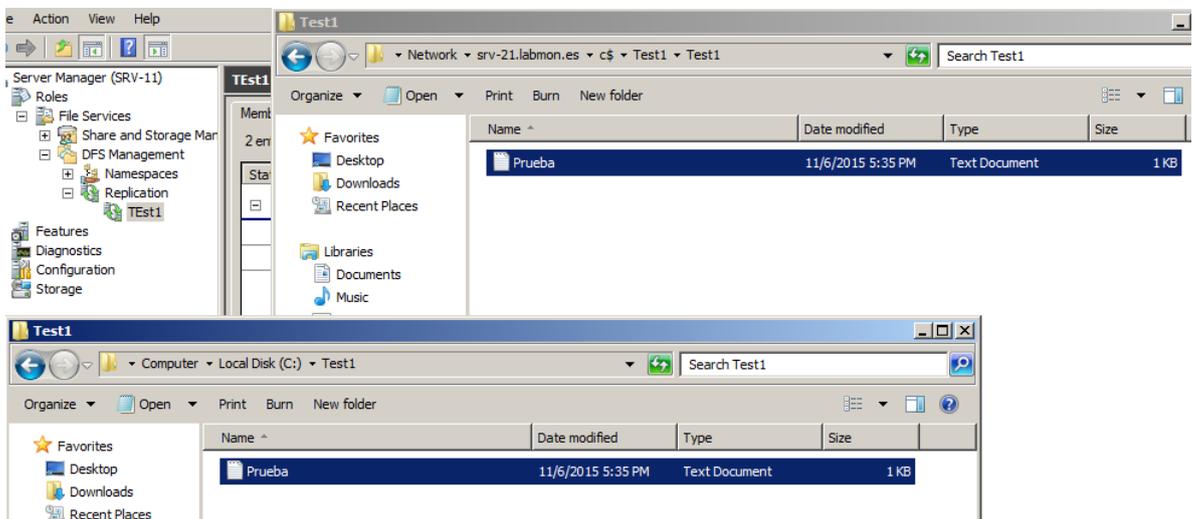


Figura 4-8: Prueba básica de replicación

4.4 Instalación y configuración de SnapMirror

Para este caso, los dos equipos tendrán una configuración idéntica (salvo nombre e IP):

NODO11	Servicios	File Server
	IP LAN	192.168.100.21/24
	Gateway	192.168.100.1
	IP iSCSI	192.168.110.21/24

Tabla 4-5: Configuración NODO11

NODO21	Servicios	File Server
	IP LAN	192.168.200.21/24
	Gateway	192.168.200.1
	IP iSCSI	192.168.210.21/24

Tabla 4-6: Configuración NODO21

Una vez que los dos equipos NetApp están instalados y configurados a nivel de sistema, lo primero que se debe de hacer es agregar la licencia de SnapMirror y habilitarlo.

```
NODO11>  
NODO11> license add AAAAAAAAAA  
license add: successfully added license key AAAAAAAAAA  
NODO11>
```

Figura 4-9: Licencia SnapMirror

```
NODO11>  
NODO11> options snapmirror.enable on  
NODO11>  
NODO11> options snapmirror  
snapmirror.access          legacy  
snapmirror.checkip.enable  off  
snapmirror.cmode.suspend  off  
snapmirror.delayed_acks.enable on  
snapmirror.enable          on  
snapmirror.log.enable      on  
snapmirror.vbn_log_enable  off  
snapmirror.volume.local_nwk_bypass.enable on  
snapmirror.vsm.volread.smtape_enable on  
NODO11>
```

Figura 4-10: Habilitar SnapMirror en el filer

Se deberán de realizar estos pasos en los dos nodos (NODO11 y NODO21).

Lo siguiente es permitir que el destino (NODO11) pueda acceder al origen (NODO21). En el origen se creará el fichero /etc/snapmirror.allow, con el nombre y la IP del destino:

```
NODO21> wfile /etc/snapmirror.allow
192.168.100.21
NODO11
read: error reading standard input: Interrupted system call
NODO21> rfile /etc/snapmirror.allow
192.168.100.21
NODO11
NODO21>
```

Figura 4-11: Permisos de acceso al filer origen

En el origen, se crea el volumen encargado de recopilar los datos, de modo restringido, lo cual permite realizar mantenimiento del volumen sin dar acceso a los datos. Esto proporciona un mayor grado de seguridad de los mismos.

```
NODO11>
NODO11>
NODO11> vol create vol_N11_bck aggr11_A 500M
Creation of volume 'vol_N11_bck' with size 500m on containing aggregate
'aggr11_A' has completed.
NODO11>
NODO11> vol restrict vol_N11_bck
Volume 'vol_N11_bck' is now restricted.
NODO11>
NODO11>
```

Figura 4-12: Creación de volumen en destino

Con estos pasos, ya estaría todo preparado para iniciar la réplica inicial, ejecutando en el nodo destino:

```
NODO11>
NODO11>
NODO11> snapmirror initialize -S 192.168.200.21:vol_N21_Org NODO11:vol_N11_bck
Transfer started.
Monitor progress with 'snapmirror status' or the snapmirror log.
NODO11>
NODO11>
```

Figura 4-13: Creación de la relación de SnapMirror entre volúmenes

De esta forma ya está configurada la réplica del volumen.

En estos momentos la réplica no tiene una configuración de periodicidad, únicamente se realizará la réplica bajo demanda. Para configurar los tiempos de réplica, es necesario crear en el destino el fichero /etc/snapmirror.conf donde especificaremos el modo y los tiempos de réplica siguiendo la siguiente estructura de fichero:

FILER_ORIGEN:VOLUMEN_ORIGEN FILER_DESTINO:VOLUMEN_DESTINO – OPCIONES MIN HORA DIA_MES DIA_SEM

Si cambiamos las opciones de días y horas por “SYNC”, la réplica se realizaría de forma síncrona.

```
NOD011>
NOD011>
NOD011> wrfile /etc/snapmirror.conf
192.168.200.21:vol_N21_Org NOD011:vol_N11_bck - sync
read: error reading standard input: Interrupted system call
NOD011>
```

Figura 4-14: Configuración en modo síncrono

```
NOD011>
NOD011>
NOD011> wrfile /etc/snapmirror.conf
192.168.200.21:vol_N21_Org NOD011:vol_N11_bck - 30 22 * *
read: error reading standard input: Interrupted system call
NOD011>
NOD011>
```

Figura 4-15: Configuración en modo programado

Para verificar el funcionamiento correcto de la réplica de volumen, se puede insertar contenido dentro del volumen de origen en el NODO21 y ver si este replica al nodo de destino.

```
NOD011>
NOD011>
NOD011> qtree status vol_N11_bck
Volume  Tree      Style Oplocks  Status
-----
vol_N11_bck          unix  enabled  normal
NOD011>
NOD011>
NOD011> qtree status vol_N11_bck
Volume  Tree      Style Oplocks  Status
-----
vol_N11_bck          unix  enabled  normal
vol_N11_bck Test1    unix  enabled  normal
NOD011>
NOD011>
NOD021>
NOD021>
NOD021> qtree status vol_N21_Org
Volume  Tree      Style Oplocks  Status
-----
vol_N21_Org          unix  enabled  normal
NOD021>
NOD021> qtree create /vol/vol_N21_Org/Test1
NOD021>
```

Figura 4-16: Verificación replica volumen (copia de Qtree)

Se realiza una nueva prueba de réplica, creando una LUN dentro del volumen origen:

```
NOD011>
NOD011> lun show
/vol/vol_N11_bck/Lun1.lun 102.0m <106928640> <r/o, online>
NOD011>
NOD011>
NOD011>
NOD021>
NOD021> lun create -s 100M -t windows_2008 /vol/vol_N21_Org/Lun1.lun
lun create: created a LUN of size: 102.0m <106928640>
NOD021>
NOD021>
```

Figura 4-17: Verificación replica volumen (copia de LUN)

Se verifica que la réplica está funcionando, dado que la Lun creada en origen se encuentra replicada en destino.

Tal y como se ha visto anteriormente, también existe la posibilidad de configurar las réplicas a nivel de Qtree. El Qtree a replicar solo será creado en el origen. En destino, se encargará SnapMirror de crearlo al iniciar la réplica:

```
NOD011>
NOD011>
NOD011> snapmirror initialize -S 192.168.200.21:/vol/vol_N21_OrgQtree/Qtree1 NOD
011:/vol/vol_N11_bckQtree/Qtree1
Transfer started.
Monitor progress with 'snapmirror status' or the snapmirror log.
NOD011>
NOD011> Fri May 20 15:14:20 GMT [NOD011:waf1.scan.ownblocks.done:info]: Complete
d block ownership calculation on volume vol_N11_bckQtree. The scanner took 10 ms
.
NOD011> Fri May 20 15:14:22 GMT [NOD011:waf1.scan.ownblocks.done:info]: Complete
d block ownership calculation on volume vol_N11_bckQtree. The scanner took 0 ms.
NOD011>
```

Figura 4-18: Creación de la relación de SnapMirror de Qtree

La mayoría de los pasos realizados para la configuración y prueba de SnapMirror, se podrían realizar a través de la herramienta de administración de NetApp "NetApp On Command System Manager".

4.5 Instalación y configuración de OpenSource de Linux

Al igual que en casos anteriores, los dos equipos tendrán una configuración similar (salvo nombre e IP):

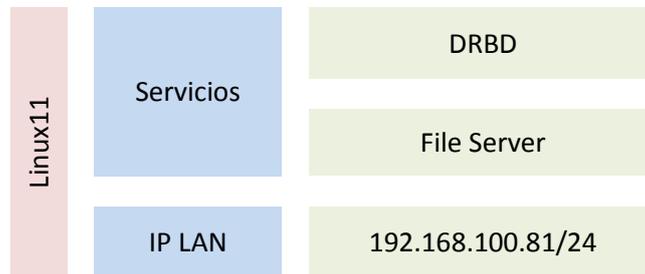


Tabla 4-7: Configuración LINUX11

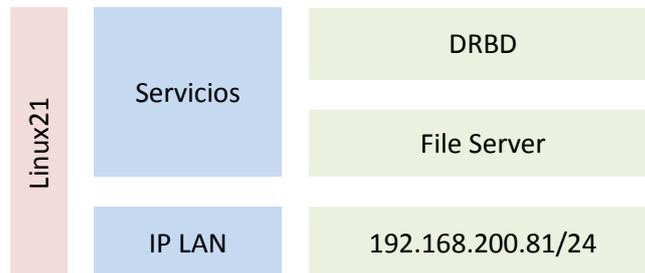


Tabla 4-8: Configuración LINUX21

En este caso se ha seleccionado Suse como distribución de Linux. Una vez que los dos equipos Linux están instalados y configurados a nivel de sistema, lo primero que se debe de hacer es agregar el módulo de “High Availability” que es donde se encuentra DRBD en Suse.

Para comenzar a utilizar DRBD, lo primero es instalar los módulos necesarios. En el caso de Suse Linux existen dos posibilidades, la primera es instalar la extensión “High Availability Extension”, la cual incluye DRBD además de otra multitud de servicios como puede ser el servicio de Cluster. LA segunda opción que existe es instalar únicamente los módulos de DRBD, que serían los siguientes:

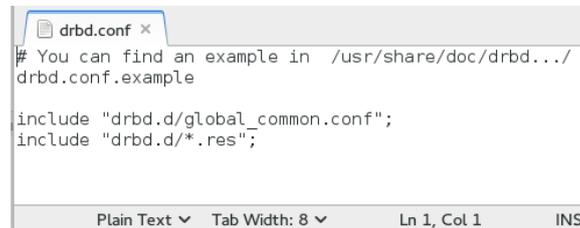
drbd	Distributed Replicated Block Device
drbd-kmp-default	Kernel driver for DRBD
drbd-kmp-xen	Kernel driver for DRBD
drbd-utils	Distributed Replicated Block Device
yast2-drbd	YaST2 – DRBD Configuration

Tabla 4-9: Módulos DRBD

Una vez terminada la instalación, y antes de poder utilizarlo, se deberá de configurar. El proceso de configuración es idéntico en ambos servidores.

Se deberá de comenzar por modificar los ficheros de configuración. El fichero `/etc/drbd.conf` deberá de contener las siguientes líneas:

```
include "drbd.d/global_common.conf";
include "drbd.d/*res";
```



```
drbd.conf x
# You can find an example in /usr/share/doc/drbd.../
drbd.conf.example

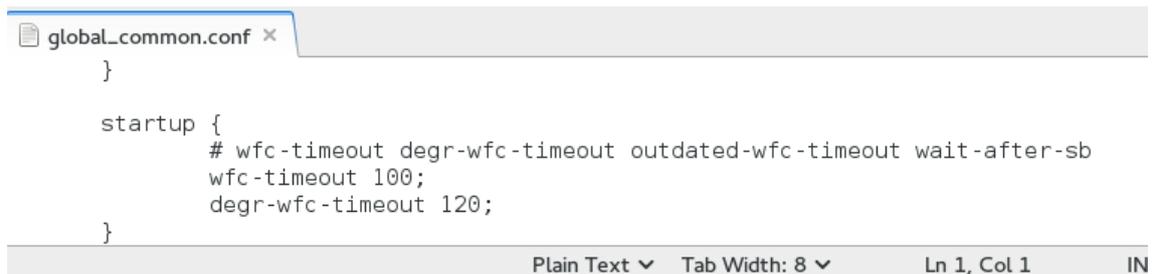
include "drbd.d/global_common.conf";
include "drbd.d/*.res";

Plain Text Tab Width: 8 Ln 1, Col 1 INS
```

Figura 4-19: Fichero `/etc/drbd.conf`

El siguiente fichero modificar es `/etc/drbd.d/global_common.conf` agregando en la parte de “startup” lo siguiente:

```
Startup {
    wfc-timeout 100;
    degr-wfc-timeout 120;
}
```



```
global_common.conf x
}

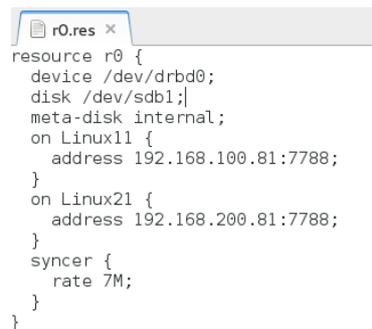
startup {
    # wfc-timeout degr-wfc-timeout outdated-wfc-timeout wait-after-sb
    wfc-timeout 100;
    degr-wfc-timeout 120;
}

Plain Text Tab Width: 8 Ln 1, Col 1 IN
```

Figura 4-20: Fichero `/etc/drbd.d/global_common.conf`

El último fichero que se necesita es /etc/drbd.dr0.res, con el siguiente contenido:

```
resource r0 {
    device /dev/drbd0;
    disk /dev/sdb1;
    meta-disk internal;
    on Linux11 {
        address 192.168.100.81:7788;
    }
    on Linux21 {
        address 192.168.200.81:7788;
    }
    syncer {
        rate 7M;
    }
}
```



```
r0.res x
resource r0 {
    device /dev/drbd0;
    disk /dev/sdb1;
    meta-disk internal;
    on Linux11 {
        address 192.168.100.81:7788;
    }
    on Linux21 {
        address 192.168.200.81:7788;
    }
    syncer {
        rate 7M;
    }
}
```

Figura 4-21: Fichero /etc/drbd.d/r0.res

Una vez modificados los ficheros en ambos servidores, es el momento de inicializar los meta-datos, para inicializarlos, es necesario verificar que el disco /dev/sdb1 está particionado y sin datos (en ambos servidores). Posteriormente ejecutamos en los dos servidores:

```
drbdadm create-md r0
rcdrbd start
```



```
Linux11:~ #
Linux11:~ # drbdadm create-md r0
--== Thank you for participating in the global usage survey ==--
The server's response is:

initializing activity log
NOT initializing bitmap
Writing meta data...
New drbd meta data block successfully created.
Linux11:~ # rcdrbd start
Linux11:~ #
```

Figura 4-22: Inicialización de meta-datos

Finalmente, iniciamos el proceso de sincronización entre servidores, ejecutando el siguiente comando en el servidor principal:

```
drbdadm -- --overwrite-data-of-peer primary r0
```

Una vez configurado, únicamente quedaría crear el sistema de ficheros y montarlo como unidad para poder utilizarlo.

5 Conclusiones

Una vez analizados los diferentes sistemas de recopilación de datos objeto de este trabajo se obtienen las siguientes lecciones aprendidas

- Los sistemas analizados son sistemas perfectamente válidos para la replicación de datos
- Los sistemas han dado unos resultados óptimos en los test realizados
- Las posibilidades de configuración de los sistemas permite tener un amplio abanico de alternativas a la hora de parametrizar la forma de implementar los diferentes sistemas tanto individualmente como entre ellos, pudiendo seleccionar uno o varios de ellos en función de las necesidades de la empresa a implementar.

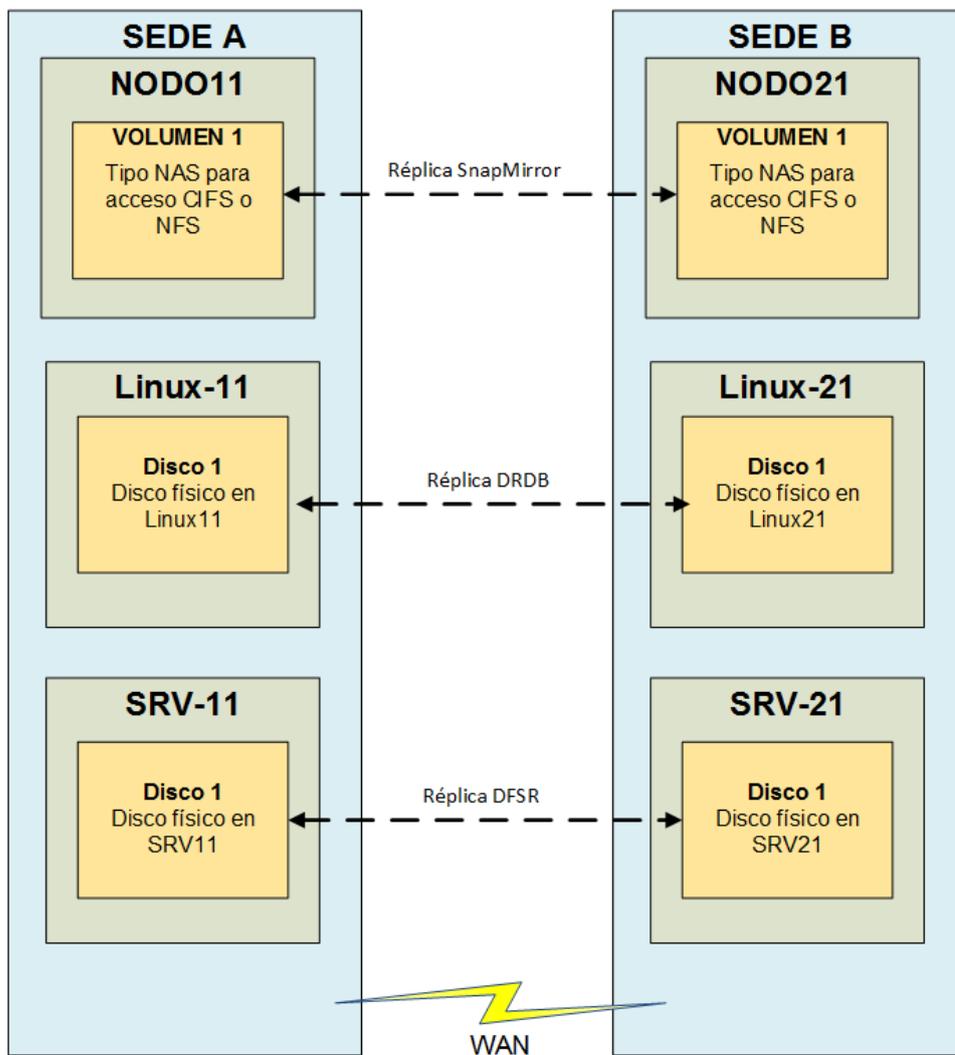


Figura 5-1: Ejemplo convivencia sistemas de réplica

- En los sistemas analizados comprobamos que DFSR y SnapMirror permiten tener una conectividad entre ellos totalmente compatibles sin que se produzcan interferencias unos con otros, sin embargo el sistema Linux solo puede interactuar con el mismo pero si puede convivir en la misma red sin interferir con los demás.

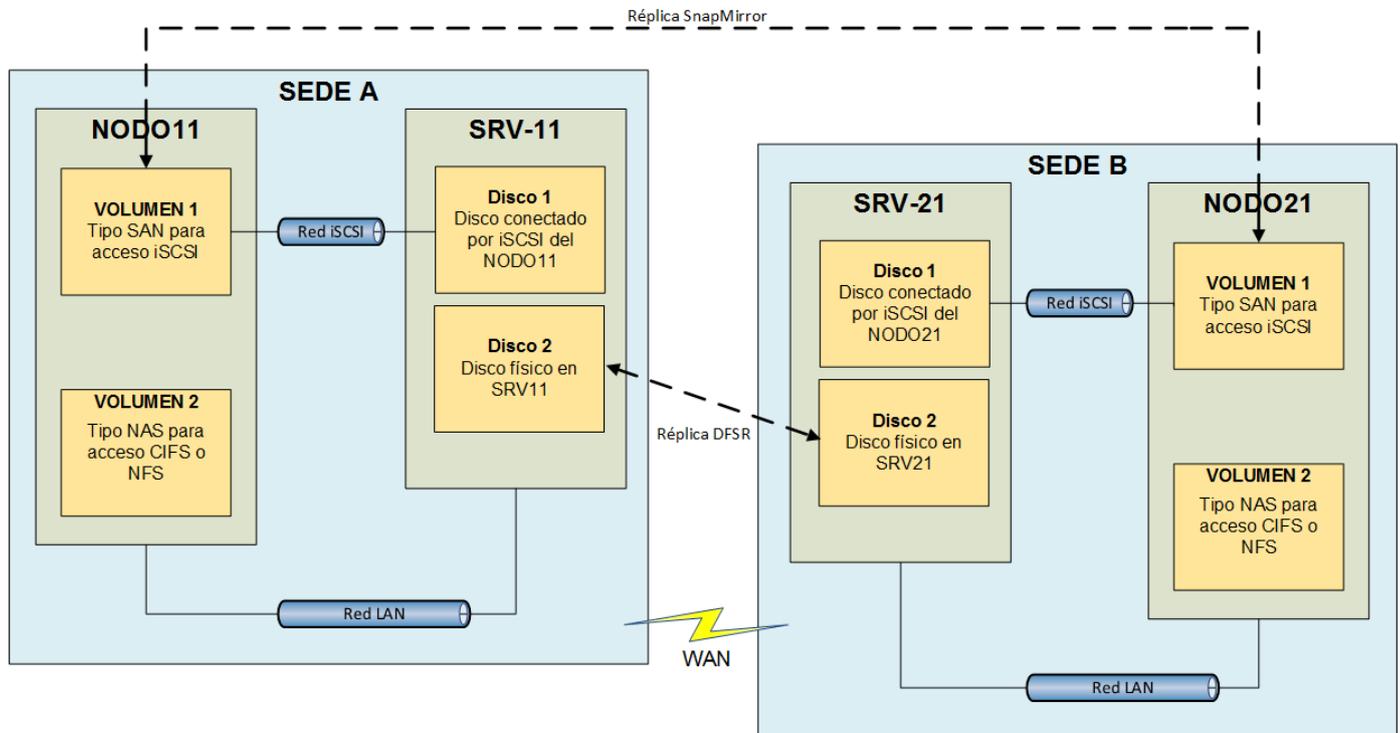


Figura 5-2: Ejemplo interacción sistemas de réplica

- Poder realizar una instalación real, tanto con un solo sistema como tener la conexión de varios sistemas intercomunicados
- Los sistemas analizados se pueden instalar en función de las necesidades de la empresa, teniendo en cuenta el nivel presupuestario del que se disponga en cada situación.

Después de finalizado el trabajo considero que los objetivos planteados inicialmente se han logrado en su totalidad. Estos objetivos eran:

- Analizar DFS-R de Microsoft.
- Analizar SnapMirror de NetApp.
- Analizar una solución Linux con software de replicación OpenSource.
- Probar y demostrar la viabilidad de los sistemas tanto de forma individual como complementándose entre sí.

La metodología y la planificación planteadas en el inicio del proyecto, se han seguido, si bien no escrupulosamente y en el orden en el que se veían reflejadas. Esto es como consecuencia a que diversas tareas no consecutivas ni encadenadas entre sí, sino que son susceptibles de solaparse unas con otras, como bien ha ocurrido en este caso, en las tareas de instalación, configuración y pruebas de los sistemas analizados.

Todo el proyecto se ha basado en situaciones teóricas sobre los productos analizados y las pruebas efectuadas han sido en un entorno de laboratorio donde los resultados obtenidos son en condiciones óptimas, sin poder reproducir escenarios reales con factores externos que pueden influir y condicionar los valores finales. Por lo tanto, el trabajo futuro que queda por explorar, es trasladar este escenario a un entorno real en producción.

6 Glosario

DFS: Distributed File System
DFS-R: Distributed File System Replication
OpenSource: Relativo al software con código abierto.
NAS: Network access server
RPO: Recovery Point Objective
RTO: Recovery Time Objective
WAN: Wide Area Network
LAN: Local Area Network
NTF: Network File System
DRBD: Distributed Replicated Block Device
RDC: Remote Differential Compression
LUN: Representación lógica de almacenamiento
Qtree: Subdirectorio dentro de la raíz de un volumen NetApp
Share: Recurso compartido.
RAID: Redundant Array of Inexpensive Disks
DNS: Domain Name Service
Qtree: Subdirectorios de un volumen raíz de NetApp

7 Bibliografía

- 1.- Blog Sistemas de Archivos Distribuidos – Febrero 2016
<http://sistemadearchivosdistribuidos03.blogspot.com.es/>
- 2.- Redmond Magazine – Febrero 2016
<https://redmondmag.com/articles/2011/02/01/dfs-best-practices.aspx>
- 3.- Technet de Microsoft – Marzo 2016
<https://technet.microsoft.com/es-es/library/jj127250.aspx>
- 4.- TechTarget – Marzo 2016
<http://searchwindowserver.techtarget.com/definition/distributed-file-system-DFS>
- 5.- My2Storage – Marzo 2016
http://www.my2storage.com/?page_id=42
- 6.- Youtube-TechEd – Marzo 2016
https://www.youtube.com/watch?v=0kPob4IKY_U
- 7.- ASIR – Marzo 2016
<http://www.asirlasgalletas.com/2016/03/xiii-dfs-sistema-de-archivos-distribuido.html>
- 8.- SysAdminsTutorials – Abril 2016
<http://www.sysadmintutorials.com/tutorials/netapp/netapp-data-ontap-7-mode/netapp-snapmirror-setup/>
- 9.- NetApp Products – Abril 2016
<http://www.netapp.com/es/products/protection-software/snapmirror.aspx>
- 10.- NetApp Tech OnTap – Abril 2016
<http://www.netapp.com/es/communities/tech-ontap/tot-btb-snapmirror-1105-es.aspx>
- 11.- NetApp Library – Abril 2016
<https://library.netapp.com/ecmdocs/ECMP1196991/html/GUID-BA1081BE-B2BB-4C6E-8A82-FB0F87AC514E.html>
- 12.- Linux-Focus – Mayo 2016
<http://es.tldp.org/LinuxFocus/pub/mirror/LinuxFocus/Castellano/March2001/article199.shtml>
- 13.- Administración de sistemas operativos – Mayo 2016
http://www.adminso.es/wiki/images/3/31/Pfc_Fransico_cap4.pdf
- 14.- DRBD a product by Linbit – Mayo 2016
<http://www.drbd.org/en/doc/users-guide-90>
http://www.drbd.org/en/doc/users-guide-90/s-distro-packages#_suse_linux_enterprise_server
- 15.- Linux OS SUSE – Mayo 2016
https://www.suse.com/documentation/sle_ha/singlehtml/book_sleha/book_sleha.html