



Clúster de alta disponibilidad y almacenamiento en NAS



Administración de Sistemas informáticos en Red
Alumnos: Luis Santana Escudero – Olga Vara Pérez
Fechas: 11-06-2018

Índice

Portada	1
Índice	2
Introducción:	4
Glosario de términos	6
Naturaleza del proyecto	8
Definición de los objetivos	8
Marco teórico	8
¿Qué son los clústeres?	8
Tipos de clúster	9
Clúster de Alta Disponibilidad	9
Componentes principales	10
Beneficios de clúster de Alta disponibilidad	10
¿Qué es un NAS?	10
¿Qué es un RAID?	11
Recursos	12
Software:	12
Apache2	12
Corosync:	13
Pacemaker:	14
FreeNAS	14
Máquinas virtuales	15
Tareas	15
Diagrama temporal	16
Configuraciones	16
Instalación Clúster de Apache	16
Instalación del Sistema Operativo	16
Instalación y preparación de Pacemaker y Corosync	18
Configuración Pacemaker y Corosync	22
NAS	24
Instalación FreeNAS	25
Configuración FreeNAS	27
Configuración carpeta compartida	28
Plan de pruebas:	36
1. Error en Corosync:	36
2. Fallo en Pacemaker:	37

3. Fallo en apache.....	37
4. Fallo del servidor:	38
Conclusiones propuestas.....	39
Bibliografía y referencias.....	40
Clúster, pacemaker y corosync:	40
NAS:	40
Configuración NFS	40
Otros:.....	40
ANEXO I – ERROR FICHERO FSTAB	41

Introducción:

En este proyecto vamos a tratar el tema de la alta disponibilidad, creando un sistema de almacenamiento compartido en un supuesto entorno empresarial.

La alta disponibilidad (High Availability), es un diseño de arquitectura de sistemas, y su implementación asociada asegura un cierto grado de continuidad operativa durante un período de medición dado, con el fin, de garantizar que el servicio funcione adecuadamente en todo momento.

Antes de comenzar con las configuraciones del sistema, comentaremos la importancia de la alta disponibilidad, ya que en el mundo empresarial existen varios servicios y aplicaciones críticas que tienen que dar un servicio continuado durante las 24 h del día todo el año.

Durante muchos años, la alta disponibilidad no ha tenido buena fama, ya que para pequeñas y medianas empresas ha sido considerada muy compleja y cara. Además se pensaba que sólo las grandes empresas, con altos fondos y múltiples recursos IT podían llevar a cabo de forma realista soluciones de alta disponibilidad. Pero con el paso de los años, y gracias a la disponibilidad de implementaciones Open Source, esta idea ha ido cambiando.

En la actualidad la alta disponibilidad combina tecnología de replicación y ritmo de los servidores para mantener sistemas informáticos que se encuentran en una ubicación remota, sincronizados con aplicaciones del centro de datos principal. Antiguamente, esto significaba la necesidad de tener redes con gran ancho de banda entre las ubicaciones físicas de las aplicaciones, así como copias de los servidores, etc. que conllevaba a que el coste de la redundancia excluyera a la alta disponibilidad del alcance de las organizaciones más pequeñas.

Hoy en día, las redes de bajo coste y gran ancho de banda están por todas partes, hasta el punto de haberse convertido en una necesidad para las empresas. Este cambio en los costes trae consigo la aparición de una gran variedad de proveedores de servicios que hace que resulte fácil crear servidores virtuales al gusto del cliente con un coste muy bajo. Estos avances en infraestructura significan que la alta disponibilidad está ahora al alcance de más organizaciones a un precio mucho más modesto.

Por lo tanto, hoy en día la alta disponibilidad tiene mucha relevancia en el mundo de las IT, y para poder medirla las empresas establecen un acuerdo a nivel de servicio (SLA) donde se definen el tiempo y el horario en el que las aplicaciones deben estar en línea. Para calcular la disponibilidad de un sistema existe una fórmula, que es la siguiente:

$$\text{Disponibilidad} = ((A - B)/A) \times 100 \text{ por ciento}$$

Índice de disponibilidad	Duración del tiempo de inactividad
97%	11 días
98%	7 días
99%	3 días y 15 h
99,9%	8 h y 48 minutos
99,99%	53 minutos
99,999%	5 minutos
99,9999%	32 segundos

Tabla 1. Alta disponibilidad

Para saber si un sistema es altamente disponible debemos comparar el porcentaje resultante en la fórmula superior con los de la tabla (Tabla 1).

Glosario de términos

- **JBOD:** JBOD (Just a Bunch Of Drives, o solo un montón de discos) no es un RAID en el puro sentido. JBOD nos crea una sola partición en varios discos (o una sola partición a partir de varias particiones), comportándose como si fuera una sola partición.
- **Licencia BSD:** es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Es una licencia de software libre permisiva. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.
- **Licencia GPL:** es la licencia de software creada para el proyecto GNU que actualmente se emplea en gran cantidad de proyectos Open Source como por ejemplo el núcleo Linux. Es una licencia de software libre que garantiza que las variaciones seguirán siendo Open Source (copyleft). La licencia GPL al contrario que la BSD no permite el uso del código fuente en software privativo.
- **Licencia Apache:** es la licencia de software creada para el proyecto Apache, que actualmente se emplea en gran cantidad de proyectos Open Source como por ejemplo el núcleo el servidor HTTP de Apache. Es una licencia de software libre permisiva. La licencia Apache, al igual que la BSD, permite ciertos usos el uso del código fuente en software privativo (no es copyleft).
- **Servidor web:** es un programa especialmente diseñado para transferir datos de hipertexto, es decir, páginas web con todos sus elementos (textos, widgets, banners, etc). Estos servidores web utilizan el protocolo HTTP.
- **Protocolo HTTP:** es el protocolo de transferencia de “hipertexto” que emplean los servidores web, y otros servicios y APIs, para comunicarse y transferir datos.

- **NFS:** es un sistema de archivos de red que permite a los hosts remotos montar carpetas sobre la red e interactuar con esos sistemas de archivos como si estuvieran montados localmente.
- **Fstab:** es un fichero que se encuentra en /etc, en él se indican los dispositivos que se deben montar, el lugar donde se monta, así como el sistema de archivos y las opciones que utilizaremos a la hora del montaje.
- **Exports:** es un fichero que se encuentra en /etc, en él se indican todos los directorios que un servidor NFS exporta a los clientes.
- **Fencing:** parte de la configuración del clúster que sirve para resetear un nodo si ha perdido conectividad, y así evitar corrupción o bloqueo en el acceso a datos.
- **Quorum:** parte de la configuración de un clúster que define que nodos están activos en el clúster y permite ubicar los servicios en ellos.
- **Runlevel:** es el modo de operación en los sistemas operativos que implementan el estilo de sistema de arranque de iniciación tipo UNIX, es decir, cuando el equipo entra al runlevel 0, está apagado, y cuando entra al runlevel 6, se reinicia. Los runlevels intermedios (1 a 5) difieren en relación con qué unidades de disco se montan, y qué servicios de red son iniciados. Los niveles más bajos se utilizan para el mantenimiento o la recuperación de emergencia, ya que por lo general no ofrecen ningún servicio de red.

Naturaleza del proyecto

El proyecto consiste en diseñar e implementar una arquitectura de alta disponibilidad, para dar un servicio continuo a los clientes. Esta arquitectura incluirá un clúster activo/activo con Apache instalado de forma activa/pasiva y un servidor NAS, accesible por NFS, donde se alojarán las páginas que servirá Apache.

La idea de la creación de este proyecto surge de nuestro interés por la asignatura de Seguridad y Alta disponibilidad. Pensamos que podría ser buena idea realizar un proyecto basado en la alta disponibilidad ya que durante el curso no profundizamos mucho en tema.

Los beneficiarios de este proyecto serán aquellas empresas que ofrezcan determinados servicios que deben estar en funcionamiento constante y, por lo tanto, que tengan que ser altamente disponibles.

Este proyecto está pensado para llevarse a cabo en un supuesto entorno empresarial donde exista la necesidad de un sistema de alta disponibilidad para el funcionamiento de determinados servicios.

Definición de los objetivos

El objetivo de nuestro proyecto es crear un sistema de alta disponibilidad para ofrecer un servicio de Apache que aloja sus páginas en un servidor NAS.

Marco teórico

¿Qué son los clústeres?

Un clúster es un tipo de sistema distribuido o paralelo conformado por una colección de computadoras interconectadas, usado como un único recurso de computación unificado.

Tipos de clúster

Los clústeres cuyos nodos tienen arquitecturas similares y un mismo sistema operativo se los conoce como Homogéneos, el caso contrario se los conoce como Heterogéneos.

Según la función que tenga el clúster, se pueden diferenciar en tres tipos:

- Clúster de Balanceo de Carga (Fig.1).
- Clúster de Alta Disponibilidad (Fig.2).
- Clúster de Alto Rendimiento (Fig.3).

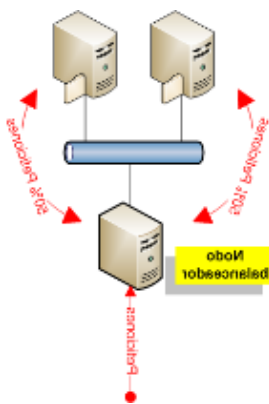


Fig. 2 Balanceo de carga

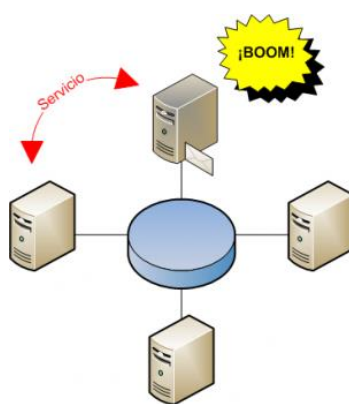


Fig. 3 Alta disponibilidad

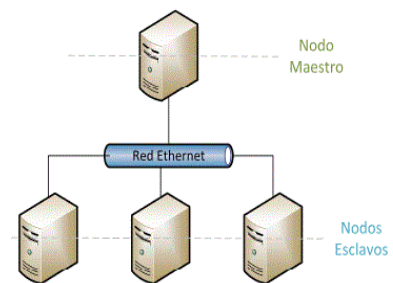


Fig. 1 Alto rendimiento

Nosotros nos centraremos en el clúster de Alta disponibilidad.

Clúster de Alta Disponibilidad

El objetivo principal de este tipo de clúster es el de mantener disponible uno o más servicios la mayor cantidad de tiempo posible.

Están caracterizados por que sus nodos están monitorizándose de manera continua, es así que, cuando se detecta un fallo de hardware o software es uno de los nodos, otro nodo se encarga de proveer los servicios del nodo con problemas, minimizando así la percepción de la pérdida de servicio por parte de los usuarios.

Componentes principales

Los componentes principales de un clúster de alta disponibilidad son:

- **Nodos:** es otro equipo dentro del clúster. No es obligatorio que todos los nodos del clúster sean igual, pero es recomendable que compartan una serie de características similares.
- **MiddleWare:** es un programa intermediario entre dos programas independientes. Es el encargado de monitorizar, administrar, distribuir y controlar el comportamiento general del clúster. En nuestro caso es Pacemaker.
- **Servicios:** son recursos que pueden ser iniciados o detenidos, en nuestro caso sería Apache.

Beneficios de clúster de Alta disponibilidad

El principal beneficio es la reducción de las interrupciones de los servicios por problemas imprevistos. Otros beneficios son el bajo costo de la operación, escalabilidad, recuperación, protección y redundancia de datos.

¿Qué es un NAS?

NAS es el nombre dado a una tecnología de almacenamiento dedicada a compartir la capacidad de almacenamiento de un ordenador (Servidor) con ordenadores personales o servidores clientes a través de una red (normalmente TCP/IP), haciendo uso de un Sistema Operativo optimizado para dar acceso con los protocolos CIFS, NFS, FTP o TFTP.

Generalmente, los sistemas NAS son dispositivos de almacenamiento específicos a los que se accede desde los equipos a través de protocolos de red (normalmente TCP/IP). También se podría considerar un sistema NAS a un servidor (Linux, Windows, ...), que comparte sus unidades por red.

Los protocolos de comunicaciones NAS están basados en ficheros por lo que el cliente solicita el fichero completo al servidor y lo maneja localmente, están por ello orientados a información almacenada en ficheros de pequeño tamaño y gran cantidad. Los protocolos usados son protocolos de compartición de ficheros como NFS, Microsoft Common Internet File System (CIFS).

Muchos sistemas NAS cuentan con uno o más dispositivos de almacenamiento para incrementar su capacidad total. Normalmente, estos dispositivos están dispuestos en RAID o contenedores de almacenamiento redundante.

Los NAS son muy útiles para proporcionar el almacenamiento centralizado a ordenadores clientes en entornos con grandes cantidades de datos. Además, puede habilitar sistemas fácilmente y con bajo costo con balance de carga, tolerancia a fallos y servidor web para proveer servicios de almacenamiento.

¿Qué es un RAID?

Las siglas RAID significan “grupo redundante de dispositivos independientes”, es decir, un sistema de almacenamiento de dato que utiliza múltiples unidades de disco entre las cuales se distribuyen y replican los datos.

Para comprender el funcionamiento de un RAID hay que tener en cuenta tres conceptos:

- **Data striping:** consiste en dividir la información en múltiples discos formando así una unidad simple y lógica de almacenamiento. Cada espacio de los discos es dividido en pequeñas partes, que se combinan de forma que la unidad lógica se compone de partes alternadas de cada disco.
- **Mirroring/espejo:** se usa en los RAID 1 y 1+0 para la recuperación de datos. Los datos se duplican en forma de espejo entre dos discos, para mantener la información en caso de que falle algún disco.

- **Paridad:** se usa en el RAID 3, 4, 5 Y 6 para la recuperación de datos. En caso de pérdida de un disco, la información de paridad se combina con la de los discos restantes para generar la información perdida.

Recursos

Durante la realización de este proyecto, podemos dividir los recursos utilizados en dos tipos: software y máquinas virtuales.

Software:

Apache2

Apache2 o Apache HTTP Server (al que nos referimos en este documento como Apache) es un servidor web HTTP, Open Source y con licencia Apache License (estas licencias están explicadas en el glosario de términos) utilizado para la distribución de páginas y servicios Web.

Es un servidor multiplataforma, es decir, que puede trabajar con diferentes sistemas operativos y a la vez mantener su rendimiento. Es gratuito, muy robusto y destaca por su seguridad y rendimiento. Las principales características de Apache son las siguientes:

- Soporte de seguridad SSL y TLS.
- Puede realizar autenticación de datos utilizando SGDB.
- Puede dar soporte a diferentes lenguajes, como Perl, PHP, Python y tcl.

Requisitos para la instalación de Apache:

Procesador	RAM	Sistema Operativo	Tamaño instalación
Pentium	64 MB	Microsoft Windows GNU/Linux MacOS	50 MB

Tabla 2 *Requisitos de Apache*

Ventajas e inconvenientes:

Apache tiene una serie de ventajas que son las siguientes:

- Instalación/Configuración. Software de código abierto.
- El servidor web Apache es completamente gratuito.
- Funcional y Soporte. Alta aceptación en la red y muy popular, esto hace que muchos programadores de todo el mundo contribuyen constantemente con mejoras, que están disponibles para cualquier persona que use el servidor web y que Apache se actualice constantemente.
- Multi-plataforma. Se puede instalar en muchos sistemas operativos, es compatible con Windows, Linux y MacOS.
- Rendimiento, tiene la capacidad de manejar más de un millón de visitas/día.
- Soporte de seguridad SSL y TLS.

Los inconvenientes son los siguientes:

- Falta de integración.
- Posee formatos de configuración NO estándar.
- No posee un buen panel de configuración.

Corosync:

Corosync es un programa de código abierto, bajo la licencia nueva BSD, el cual es utilizado para desarrollar, lanzar y dar soporte a un clúster.

Corosync ofrece una serie de características adicionales para implementar la alta disponibilidad, sin la necesidad de instalar otras aplicaciones. Proporciona capacidades de membresía y mensajería de clúster, a menudo denominada capa de mensajería, a los servidores del cliente.

Por lo tanto, Corosync permite que los servidores se comuniquen como un clúster.

Pacemaker:

Pacemaker es un administrador de recursos de clúster de código abierto, es un sistema que coordina los recursos y servicios que se administran y se ponen a disposición de un clúster.

Por lo tanto, Pacemaker brinda la capacidad de controlar cómo se comporta el clúster.

Logra la máxima disponibilidad para sus servicios de clúster mediante la detección y recuperación de fallos a nivel de nodos y recursos mediante el uso de las capacidades de mensajería y membresía proporcionadas por Corosync.

Puede hacerlo para clústeres de prácticamente cualquier tamaño y viene con gran modelo de dependencia que permite al administrador expresar con precisión las relaciones (tanto de ordenamiento como de ubicación) entre los recursos del clúster.

FreeNAS

FreeNAS es un sistema operativo basado en FreeBSD que nos permite utilizar nuestro ordenador como si fuera un NAS, de manera sencilla gracias a la interfaz web que ofrece.

Se puede instalar en un disco duro, en una llave USB o arrancarlo directamente desde un LiveCD. Nosotros lo instalaremos en una Máquina virtual de VirtualBox.

FreeNAS tiene multitud de ventajas, una de ellas son la serie de servicios adicionales que nos puede prestar, ya que dispone de soporte para protocolos FTP, NFS, TFTP, UPnP o iTunes/DAAP, que permiten ampliar la funcionalidad del disco duro de red. Además, cuenta con un cliente torrent para gestionar las descargas o un servidor WEB, convirtiéndose así en algo más complejo que un NAS.

Por una parte, puede tener distintas configuraciones ya sean en RAID 0,1 o 5 o de JBOD, dependiendo de nuestras necesidades, ya sean aportar más seguridad al almacenamiento o aumentar su capacidad.

Por otra parte, dispone de copias de seguridad para la configuración del sistema, de esta forma, si hacemos algún cambio que provoque cualquier error tendrías la opción de volver atrás. Con respecto a la seguridad, nos permite cifrar el contenido de los discos duros y las comunicaciones de la transferencia de los archivos, además, de crear usuarios y grupos con distintos privilegios para llevar la gestión del NAS.

Máquinas virtuales

Nombre	Memoria	Disco duro	Tamaño de discos	Nº Tarjetas	Tipo Red
NAS	8GB	Reservado dinámicamente	- Disco principal: 8GB - RAID5: 3 discos de 16GB	1	Adaptador Puente
Server	2GB	Reservado dinámicamente	10 GB	2	Red interna
Server2	2GB	Reservado dinámicamente	10 GB	2	Red interna

Tabla 3 Máquinas virtuales

Tareas

- Creación máquinas virtuales.
- Configuración del clúster.
- Configuración NAS.
- Configuración Apache.

Diagrama temporal

TAREAS SEMANAS	1	2	3	4	5	6	7
Creación y configuración del clúster							
Creación y configuración del NAS							
Preparación del documento							
Power Point							
Presentación							

Tabla 4 Diagrama temporal

Configuraciones

Instalación Clúster de Apache

Los requisitos para la instalación de los servidores Apache son los siguientes:

- Dos máquinas virtuales.
- Servidor de Linux como sistema operativo (64 bit).
- Instalación pacemaker, corosync y apache2 en ambos nodos.

Instalación del Sistema Operativo.

Una vez que tengamos ambas máquinas virtuales creadas, las iniciamos para poder comenzar con la instalación del Sistema Operativo. El primer paso es elegir el idioma que vamos a utilizar (Fig.5), en nuestro caso el español, la ubicación para que se configure automáticamente la hora (Fig.4), y la disposición del teclado.



Fig. 5 Idioma

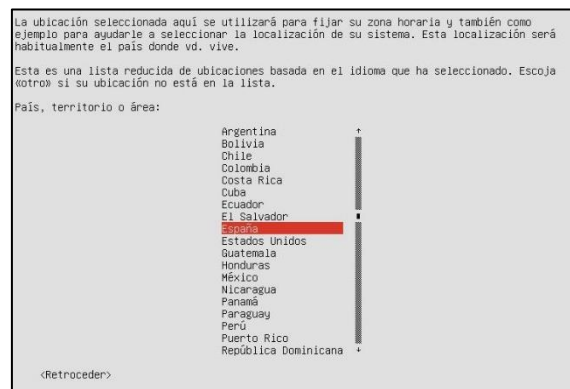


Fig. 4 Ubicación

Tras estas configuraciones comenzará la descarga de los componentes adicionales que utiliza Ubuntu.

También debemos configurar el usuario que utilizaremos como administrador del sistema (Fig.7) y su contraseña (Fig.6). El nombre del usuario será server, para el nodo1 y server2 para el nodo2.

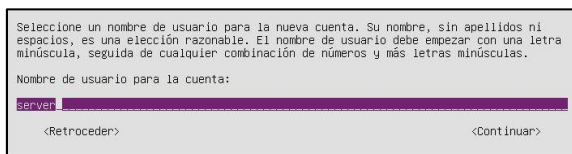


Fig. 7 Usuario

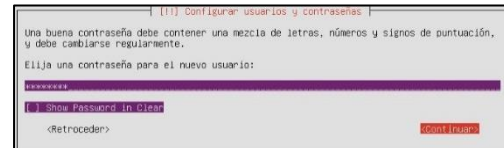


Fig. 6 Contraseña

El siguiente paso es configurar los discos duros, utilizaremos el que hemos creado al configurar las máquinas virtuales (Fig.8 y 9). Y con respecto a la administración de actualizaciones, seleccionaremos la opción que dice que se instalen automáticamente.

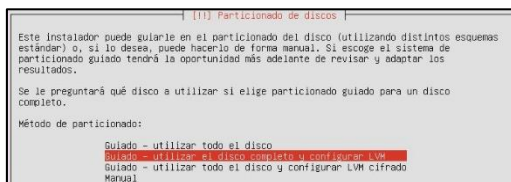


Fig. 9 Disco duro



Fig. 8 Disco duro

Por último, seleccionamos los programas que necesitamos, en nuestro caso ninguno, (Fig.11) e instalamos el cargador de arranque GRUB (Fig.12).



Fig. 11 Funcionalidades del sistema

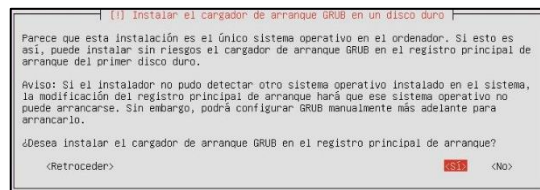


Fig. 10 Cargador de arranque GRUB

Instalación y preparación de Pacemaker y Corosync

El primer paso es instalar el software correspondiente en ambos nodos. Este paso se puede hacer de dos formas, descargando el código fuente y compilándolo (sería más compleja ya que pacemaker tiene una gran multitud de dependencias que también habría que compilar) y la otra es con el gestor de paquetes apt-get, que es la que usaremos.

El comando utilizado es el siguiente (Fig.12):

```
sudo apt-get install pacemaker corosync apache2
```

```
server@server:~$ sudo apt install pacemaker corosync apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
apache2 ya está en su versión más reciente (2.4.18-2ubuntu3.8).
Se instalarán los siguientes paquetes adicionales:
```

Fig. 12 Instalación software

Para que cada nodo forme parte del clúster deben compartir la misma clave de autenticación. Esto es una medida de seguridad para evitar que, si es descubierto el clúster, alguna persona pueda añadir los nodos que quiera, con algún fin malicioso. Para compartir la clave hacemos lo siguiente:

1. Creamos la clave en el nodo1 con el comando (Fig.13) `corosync-keygen`.

Esta clave se crea en el directorio `/etc/corosync/authkey`

```
server@server:/etc/corosync$ sudo corosync-keygen
Corosync Cluster Engine Authentication key generator.
Gathering 1024 bits for key from /dev/random.
Press keys on your keyboard to generate entropy.
Press keys on your keyboard to generate entropy (bits = 152).
Press keys on your keyboard to generate entropy (bits = 200).
Press keys on your keyboard to generate entropy (bits = 248).
Press keys on your keyboard to generate entropy (bits = 296).
Press keys on your keyboard to generate entropy (bits = 344).
Press keys on your keyboard to generate entropy (bits = 392).
Press keys on your keyboard to generate entropy (bits = 440).
Press keys on your keyboard to generate entropy (bits = 488).
Press keys on your keyboard to generate entropy (bits = 536).
Press keys on your keyboard to generate entropy (bits = 584).
Press keys on your keyboard to generate entropy (bits = 632).
Press keys on your keyboard to generate entropy (bits = 680).
Press keys on your keyboard to generate entropy (bits = 728).
Press keys on your keyboard to generate entropy (bits = 776).
Press keys on your keyboard to generate entropy (bits = 824).
Press keys on your keyboard to generate entropy (bits = 872).
Press keys on your keyboard to generate entropy (bits = 920).
Press keys on your keyboard to generate entropy (bits = 968).
Press keys on your keyboard to generate entropy (bits = 1016).
Writing corosync key to /etc/corosync/authkey.
```

Fig. 13 Creación de clave

2. Copiamos la clave al nodo2 utilizando el comando (Fig.14):

```
scp clave_nodo1 IP_nodo2:carpeta_donde_lo_copiamos_nodo2
```

```
server@servidor-1:/etc/corosync$ sudo scp /etc/corosync/authkey 192.168.3.103:/etc/corosync/
root@192.168.3.103's password: █
```

Fig. 14 Compartir claves

Comprobamos que se ha copiado correctamente en el nodo2 (Fig.15).

```
server2@server:~$ ls
authkey  Documentos  Imágenes  Plantillas  Videos
Descargas  Escritorio  Música    Público
server2@server:~$ sudo mv authkey /etc/corosync/
server2@server:~$ ls
Descargas  Escritorio  Música    Público
Documentos Imágenes   Plantillas Videos
server2@server:~$ ls /etc/corosync/
authkey corosync.conf
server2@server:~$ █
```

Fig. 15 Comprobación de clave

3. Ahora debemos dar permisos a la clave, ya que si no los tiene no iniciará. Para ello hemos usado el comando (Fig.16) `chmod 400` (sólo lectura para el usuario root) `/etc/corosync/authkey`.

```
server2@server:~$ sudo chmod 400 /etc/corosync/authkey
server2@server:~$ █
```

Fig. 16 Permisos a la clave

- Para especificar la red en la que se va a escuchar el heartbeat de Pacemaker, editaremos el fichero `/etc/corosync/corosync.conf`.

Hay que descomentar la fila de `bindnetaddr` y poner la red virtual que hemos creado (Fig.17).

```
interface {
    # Rings must be consecutively numbered, starting at 0.
    ringnumber: 0
    # This is normally the *network* address of the
    # interface to bind to. This ensures that you can use
    # identical instances of this configuration file
    # across all your cluster nodes, without having to
    # modify this option.
    bindnetaddr: 192.168.2.0
    # However, if you have multiple physical network
    # interfaces configured for the same subnet, then the
    # network address alone is not sufficient to identify
    # the interface Corosync should bind to. In that case,
    # configure the *host* address of the interface
```

Fig. 17 Fichero `/etc/corosync/corosync.conf`

- Por último, reiniciamos con el comando (Fig.18):

```
service corosync restart
```

```
server@server:/etc/corosync$ sudo service corosync restart
```

Fig. 18 Reinicio del servicio

Para que el servicio corosync se inicie al iniciar el sistema debemos editar o añadir la fila `START` del fichero `/etc/default/corosync` y poner `START="yes"` (Fig.19):

```
GNU nano 2.5.3      Archivo: /etc/default/corosync
# Corosync runtime directory
#COROSYNC_RUN_DIR=/var/lib/corosync

# Path to corosync.conf
#COROSYNC_MAIN_CONFIG_FILE=/etc/corosync/corosync.conf

# Path to authfile
#COROSYNC_TOTEM_AUTHKEY_FILE=/etc/corosync/authkey

# Command line options
#OPTIONS=""

START="yes"
```

Fig. 19 Fichero `/etc/default/corosync`

Configuración de la interfaz virtual de red

En este apartado explicaremos como hemos creado la interfaz virtual que es la que realmente recibe las peticiones del cliente. La configuración de red es la siguiente (Tabla.5):

Nodo	Interfaz	IP
Nodo1	Enp0s8	192.168.2.104
Nodo2	Enp0s8	192.168.2.105
Interfaz virtual	Eth2:0	192.168.2.101

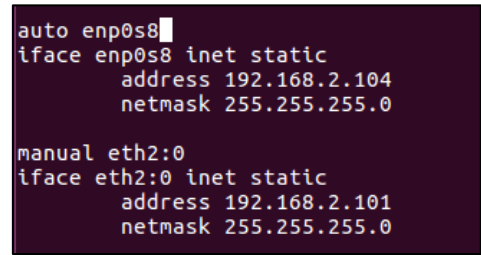
Tabla 5 Configuración de red

Para crear la interfaz virtual, debemos editar el fichero `/etc/network/interfaces` abriéndolo con nano en ambos nodos y añadirla (Fig.20):

Para añadirla hay que poner:

`manual eth2:0` (en nuestro caso)

```
iface eth2:0 inet static
    address 192.168.0.101
    netmask 255.255.255.0
```



```
auto enp0s8
iface enp0s8 inet static
    address 192.168.2.104
    netmask 255.255.255.0

manual eth2:0
iface eth2:0 inet static
    address 192.168.2.101
    netmask 255.255.255.0
```

Fig. 20 Fichero `/etc/network/interfaces`

Para asegurar que la red se ha configurado correctamente, reiniciamos (Fig.21) el fichero de red `/etc/init.d/networking restart`



```
server2@server:~$ sudo /etc/init.d/networking restart
[ ok ] Restarting networking (via systemctl): networking.service.
server2@server:~$
```

Fig. 21 Reinicio interfaz de red

Cuando tengamos las tres redes listas, podremos comenzar con la configuración de Pacemaker y corosync.

Configuración Pacemaker y Corosync

Interfaz virtual en alta disponibilidad

La función de la interfaz virtual es recibir las peticiones del cliente cuando el nodo activo caiga, de manera que el servicio este siempre disponible y se traspase al segundo nodo de forma transparente a los clientes.

Los pasos a seguir se deben realizar en el nodo1 y son los siguientes:

1. Desactivar stonith, que se utiliza para parar nodos problemáticos de forma automática. También debemos desactivar quorum, que sirve para determinar cuál será el nodo que se pondrá en funcionamiento en el caso de que haya algún problema en el nodo activo, en nuestro caso no es necesario ya que sólo tendremos dos nodos (Fig.22).

```
root@server:/etc/corosync# crm configure property stonith-enabled=false
root@server:/etc/corosync# crm configure property no-quorum-policy=ignore
```

Fig. 22 Desactivar stonith

2. Añadimos la interfaz virtual (Fig.23) que hemos creado anteriormente, ya que a través de ella se producirá el heartbeat que une ambos nodos.

```
root@server:/etc/corosync# crm configure primitive FAILOVER-ADDR ocf:heartbeat:IPaddr2 params ip="192.168.2.101" nic="enp0s8" op monitor interval="10s" meta is-managed="true"
```

Fig. 23 Interfaz virtual

Para comprobar que el clúster funciona correctamente realizaremos la siguiente prueba.

Primero, comprobamos el estado con el comando `crm_mon` 

Si observamos la imagen (Fig.24) nos encontramos que los dos nodos están activos.

```

Last updated: Wed May 16 15:39:46 2018      Last change: Wed May 16 15:39:16
2018 by root via cibadmin on server
Stack: corosync
Current DC: server (version 1.1.14-70404b0) - partition with quorum
2 nodes and 1 resource configured

Online: [ server server2 ]

FAILOVER-ADDR (ocf::heartbeat:IPaddr2):      Started server

```

Fig. 24 Estado del clúster

Ahora pararemos el corosync, con el comando `# service corosync stop` y volveremos a comprobar el estado, en este caso el nodo que está en funcionamiento es el nodo2, que ha asumido la carga de trabajo que recibía el nodo1 (Fig.25).

```

Last updated: Wed May 16 15:55:24 2018      Last change: Wed May 16 15:38:43
2018 by root via cibadmin on server
Stack: corosync
Current DC: server2 (version 1.1.14-70404b0) - partition WITHOUT quorum
2 nodes and 1 resource configured

Online: [ server2 ]
OFFLINE: [ server ]

FAILOVER-ADDR (ocf::heartbeat:IPaddr2):      Started server2

```

Fig. 25 Estado del clúster

Una vez que hayamos comprobado el correcto funcionamiento de ambos nodos, volvemos a levantar el servicio con el comando `# service corosync start` y reiniciar el pacemaker con el comando `# service pacemaker restart`.

Configuración Apache

Apache será el servicio que tendrá alta disponibilidad gracias al clúster. Anteriormente lo habíamos instalado, pero aún no estaba configurado dentro del clúster, para ello ejecutaremos el siguiente comando (Fig.26) `# crm configure primitive FAILOVER-APACHE lsb:apache2 op monitor Interval="15"` y para una mejor organización agruparemos las dos configuraciones (heartbeat y apache) en un

mismo grupo, para ello usaremos el comando (Fig.26): `# crm configure group mi_cluster_web FAILOVER-ADDR FAILOVER-APACHE` como se muestra en la siguiente captura:

```
root@server:/etc/corosync# crm configure primitive FAILOVER-APACHE lsb::apache2
op monitor interval="15s"
root@server:/etc/corosync# crm configure group mi_cluster_web FAILOVER-ADDR FAILOVER-APACHE
```

Fig. 26 Configuración de Apache en el clúster

Ya está añadido al clúster, ahora comprobaremos que funciona correctamente realizando las mismas pruebas que hemos realizado anteriormente.

Primero comprobamos el estado del clúster con el comando `crm_mon`, podemos observar que se ha añadido correctamente la nueva configuración y ambas configuraciones se encuentran situadas en el mismo grupo (Fig.27).

```
Last updated: Thu May 17 09:14:50 2018      Last change: Thu May 17 09:13:44
2018 by root via cibadmin on server
Stack: corosync
Current DC: server (version 1.1.14-70404b0) - partition with quorum
2 nodes and 2 resources configured

Online: [ server server2 ]

Resource Group: mi_cluster_web
  FAILOVER-ADDR      (ocf::heartbeat:IPaddr2):      Started server
  FAILOVER-APACHE    (lsb::apache2):      Started server
```

Fig. 27 Estado del clúster

NAS

Como hemos explicado anteriormente, para la configuración del NAS utilizaremos el software FreeNAS que aporta una serie de servicios adicionales.

FreeNAS es un sistema operativo libre que podemos encontrar en su propia web. Nosotros nos hemos descargado la versión 11. La ISO ocupa unos 604Mb.

Una vez que hayamos descargado la ISO, creamos la máquina virtual que va a hacer de NAS con los siguientes requisitos:

- Se requiere hardware de 64 bits para las versiones actuales de FreeNAS. FreeNAS 9.2.1.9 fue la última versión que admitió hardware de 32 bits y sistemas de archivos UFS.
- Los SSD, SATADOM o memorias USB pueden usarse para dispositivos de arranque. Se recomiendan SSD.
- 8 GB de RAM es el requisito mínimo absoluto. 1 GB por terabyte de almacenamiento es un punto de partida estándar para calcular las necesidades adicionales de RAM, aunque las necesidades reales varían.
- Los discos de almacenamiento conectados directamente son necesarios para que FreeNAS proporcione tolerancia a fallos. No se recomiendan tarjetas RAID de hardware porque impiden este acceso directo y reducen la confiabilidad.
- Se recomiendan discos duros NAS específicos como WD Red.
- Se recomiendan las tarjetas Intel o Chelsio 1 GbE o 10 GbE Ethernet.

Cuando tengamos la máquina virtual, añadiremos una controladora SATA y 3 discos duros, ya que utilizaremos una configuración en RAID5, asegurando la redundancia de los datos, para que si uno falla no haya pérdida de datos, aunque esto provoque la pérdida de un tercio de cada disco.

Instalación FreeNAS

Al encender la máquina virtual, se lanza la instalación y aparece la siguiente ventana (Fig.28), se debe seleccionar la primera opción.

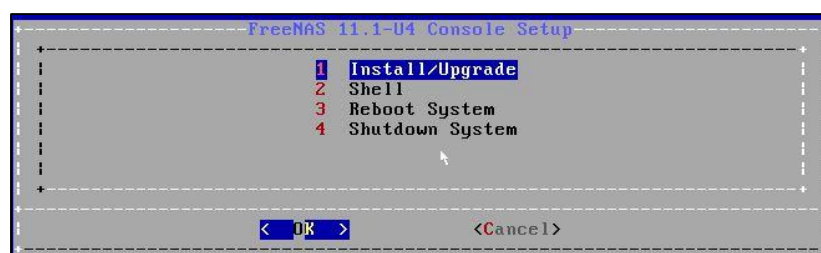


Fig. 28 Instalación FreeNAS

A continuación, seleccionamos el disco duro donde lo vamos a instalar, aceptamos y aparecerá un mensaje de advertencia que indica que el disco no se podrá utilizar, ya que sólo será utilizado para albergar el sistema operativo (Fig.29).

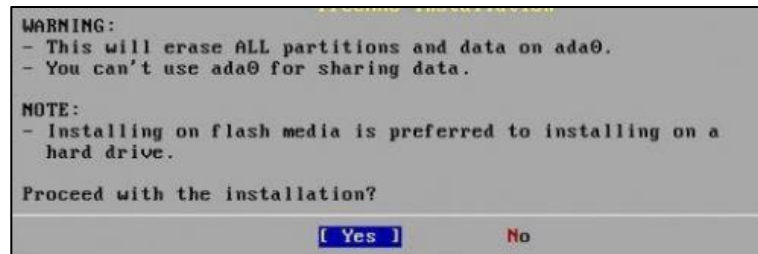


Fig. 29 Advertencia

Una vez que haya terminado la instalación, debemos clicar en OK y aparecerá la pantalla inicial de instalación, donde debemos seleccionar la 11ª opción para apagar la máquina virtual, extraer el CD y ampliar la memoria. Tras esto reiniciamos y aparecerá la siguiente pantalla (Fig.30).

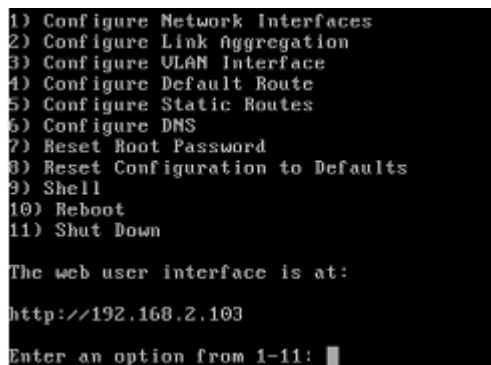


Fig. 30 Interfaz NAS

Para acceder a FreeNAS, debemos utilizar la IP que nos proporciona, en nuestro caso <http://192.168.2.103> y entrar desde el navegador.

Para que funcione dicho acceso web, la máquina no puede estar como NAT.

La interfaz que veremos desde la web es la siguiente (Fig.31) , nos pide nuestro usuario y contraseña, los añadiremos y entraremos en la web.

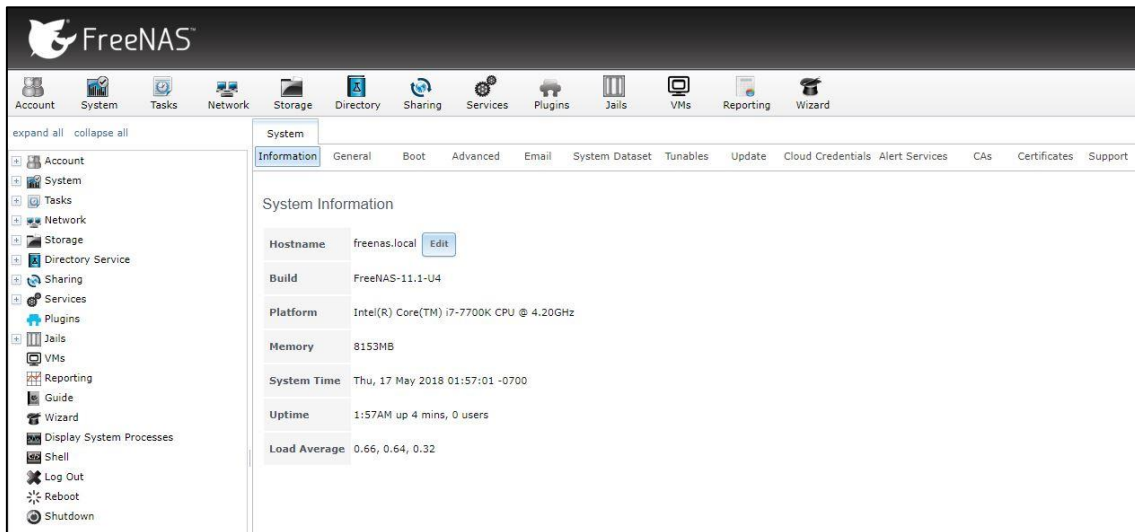


Fig. 31 FreeNAS web

Configuración FreeNAS

En primer lugar, editaremos la información general del NAS. Para ello nos situaremos en la pestaña general, del apartado sistema (Fig.32):

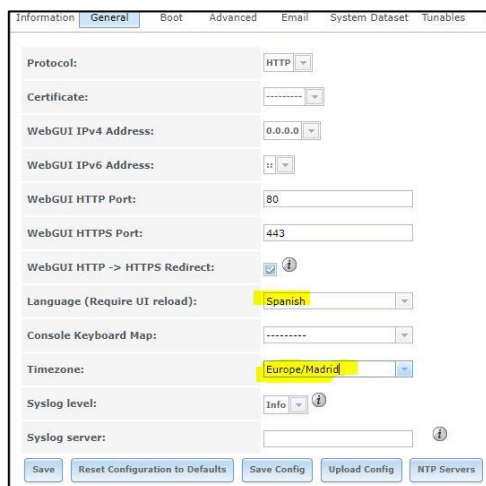


Fig. 32 Información general NAS

Lo único que modificaremos es el apartado WebGUI IPv4 Address donde pondremos la IP por la que se accederá a la web, que en nuestro caso es la 192.168.2.103.

Además, modificamos el lenguaje y la zona horaria a español y Europa/Madrid.

Cuando ya tenemos configurada toda la información general, pasaremos a crear el volumen donde estarán almacenadas las páginas de Apache. Para crearlo, nos situaremos en el menú de la izquierda y desplegamos la sección Almacenamiento -> Volúmenes y pulsamos sobre Volume Manager.

Añadimos la siguiente información (Fig.33) :

- El nombre del volumen, en nuestro caso web.
- Se puede encriptar o no, nosotros no lo hemos encriptado.
- Discos disponibles, seleccionamos el único que tenemos con 17 GB.
- El volumen será distribuido en raid-z.

Por último, clicamos sobre añadir el volumen y en el menú de la izquierda en la parte de volúmenes, debería aparecer el que hemos creado, tal y como se muestra en la (Fig.34).

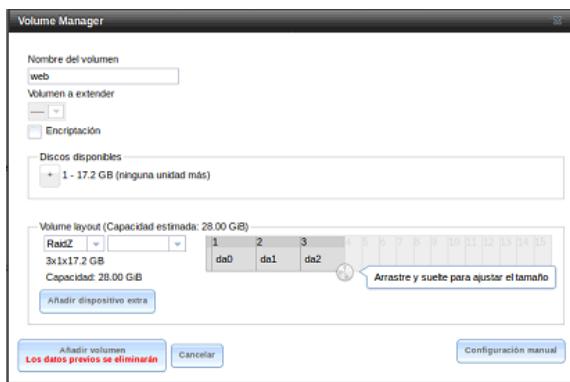


Fig. 33 Creación volumen



Fig. 34 Volumen creado

Configuración carpeta compartida

En este apartado, explicamos los pasos que hay que seguir para compartir el volumen del NAS con el clúster mediante el protocolo NFS.

Configuración de carpeta compartida por NFS

El primer paso, es habilitar el servicio de NFS en el NAS. Para ello, debemos ir a servicios -> control de servicios situado en el menú de la izquierda (Fig.35) . Lo habilitamos clicando sobre el botón rojo que pone “stopped”, ahora debería aparecer en color verde (Fig.36) .



Fig. 35 Habilitar NFS

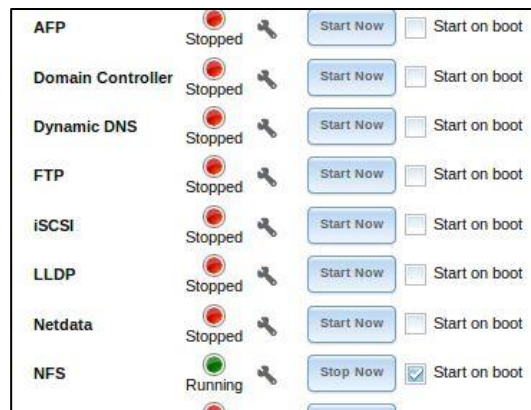


Fig. 36 Habilitar NFS

Para configurar la dirección IP que está enlazada con ese servicio y el número de servidores que lo van a utilizar, clicamos sobre la llave inglesa y se abrirá una ventana como la siguiente (Fig.37). Nosotros pondremos la IP que le hemos asignado al NAS, que es la 192.168.2.103

Además, podemos configurar otra serie de opciones que nosotros no hemos tocado. Como por ejemplo habilitar NFSv4...

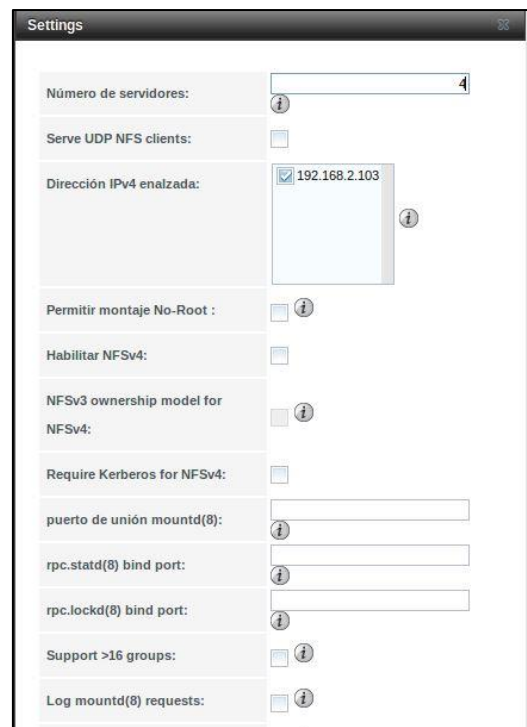


Fig. 37 Configuración NFS



Fig. 38 Configuración NFS

Una vez que lo hemos habilitado, configuraremos el volumen creado anteriormente como compartido NFS. Vamos a Compartiendo -> Compartido Unix (NFS) -> Añadir compartidos NFS (Unix) (Fig.38) .

Al clicar, nos aparece una ventana (Fig.39) donde tenemos que seleccionar la ruta del volumen que vamos a compartir por NFS, en nuestro caso /mnt/web, la red que utilizaremos que es la 192.168.2.0/24 (hay que añadirla con el mismo formato), las IPs que permitiremos que accedan al volumen, en nuestro caso las de los nodos del clúster 192.168.2.104,105. Y, por último, seleccionaremos el usuario que tendrá permisos de lectura sobre el volumen, en nuestro caso root. También se puede configurar, el grupo del usuario, todos los usuarios y todos los grupos. Nosotros eso lo hemos dejado vacío.

The screenshot shows a configuration window for NFS. At the top, there is a 'Ruta:' field with the value '/mnt/web/prueba' and a 'Navegar' button. Below it is a 'Borrar:' checkbox. A section titled 'Add extra Ruta' contains a 'Comentario:' field. The 'Redes Autorizadas:' section has a text area with '192.168.2.0/24' and a list of IP addresses '192.168.2.104 192.168.2.105'. Below this is a note: 'Este es el puerto donde WebDAV esta ejecutandose.
No use un puerto que ya se encuentra en uso por otro servicio (ej: 22 para SSH)'. There are checkboxes for 'Todos los directorios:', 'Solo Lectura:', and 'Silencio:'. The 'Mapa raíz de Usuario:' dropdown is set to 'root'. Other dropdowns for 'Mapa raíz de Grupo:', 'Mapa de todos los Usuarios:', and 'Mapa de todos los grupos:' are empty. At the bottom are buttons for 'OK', 'Cancelar', 'Borrar', and 'Modo Básico'.

Fig. 39 Configuración NAS

Hasta aquí ya tendríamos configurado el servicio NFS en el NAS. Ahora realizaremos las configuraciones necesarias en los nodos para que puedan acceder, los pasos se deben hacer en ambos nodos:

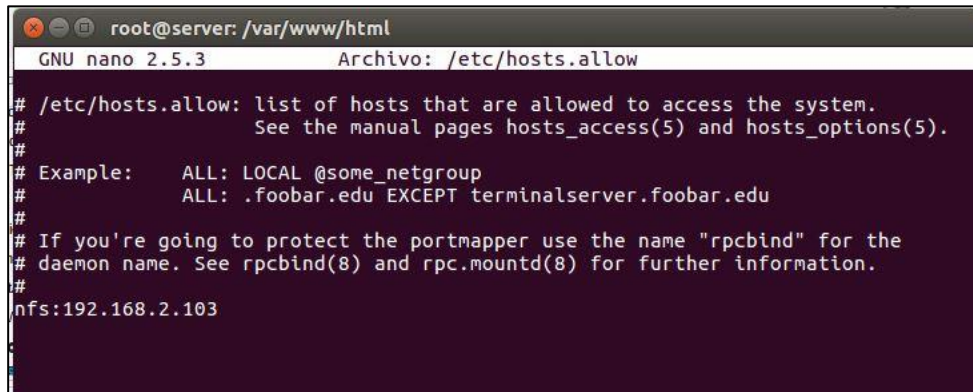
1. En primer lugar, instalamos el paquete NFS para poder utilizar el protocolo a transferir los ficheros (Fig.40) . Utilizamos el comando `apt-get install nfs-common`,

```
root@server:/var/www/html# apt-get install nfs-common
```

Fig. 40 Instalación NFS

2. En segundo lugar, editamos el archivo `/etc/hosts.allow` para añadir la IP del NAS, que es la que va a transferir por NFS (Fig.41) . Añadimos la línea
`nfs:192.168.2.103`

Este fichero sirve para permitir el acceso a los servicios del sistema.



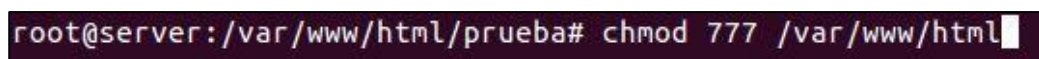
```

root@server: /var/www/html
GNU nano 2.5.3 Archivo: /etc/hosts.allow
# /etc/hosts.allow: list of hosts that are allowed to access the system.
# See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:  ALL: LOCAL @some_netgroup
#           ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#
nfs:192.168.2.103

```

Fig. 41 Fichero `hosts.allow`

3. En tercer lugar, elegimos cual será la carpeta que montaremos del clúster, en nuestro caso es `/var/www/html`, ya es en esta donde se guardan las páginas de Apache. Además, cambiaremos los permisos para posteriormente poder crear páginas y verlas (Fig.42) . Utilizamos el comando `# chmod 777 /var/www/html` para que el usuario del servidor tenga control total.



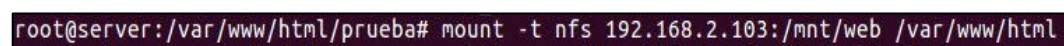
```

root@server:/var/www/html/prueba# chmod 777 /var/www/html

```

Fig. 42 Cambio de permisos

4. En cuarto lugar, montaremos el directorio de red NFS (Ip del NAS:volumen del NAS) sobre el directorio local (`/var/www/html`). Para ello utilizaremos el comando `# mount -t nas 192.168.2.103:/mnt/web /var/www/html`. Lo único a destacar de este comando es el parámetro `-t` que indica el tipo de transferencia de fichero, en nuestro caso NFS (Fig.43).



```

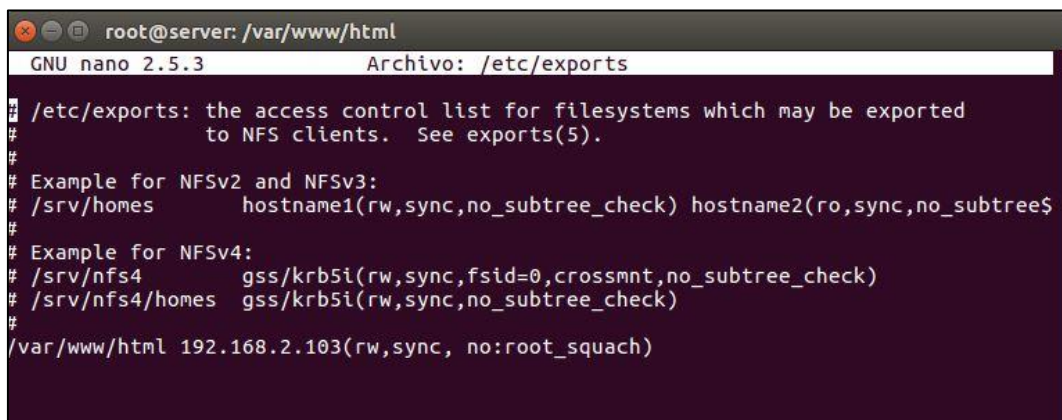
root@server:/var/www/html/prueba# mount -t nfs 192.168.2.103:/mnt/web /var/www/html

```

Fig. 43 Montaje del volumen

5. En quinto lugar, editamos el fichero `/etc/exports` donde indicaremos la carpeta que se va a exportar y la máquina destino con los permisos y opciones necesarias (Fig.44).

Añadiremos la línea `/var/www/html 192.168.2.103(rw,sync, no:root_squach)` donde primero ponemos el directorio a compartir, después la IP seguida de las opciones que deseemos, en nuestro caso le asignamos permisos de lectura y escritura, sincronización y para que root tenga el control.



```

root@server: /var/www/html
GNU nano 2.5.3 Archivo: /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree$
#
# Example for NFSv4:
# /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
#
/var/www/html 192.168.2.103(rw,sync, no:root_squach)

```

Fig. 44 Fichero `/etc/exports`

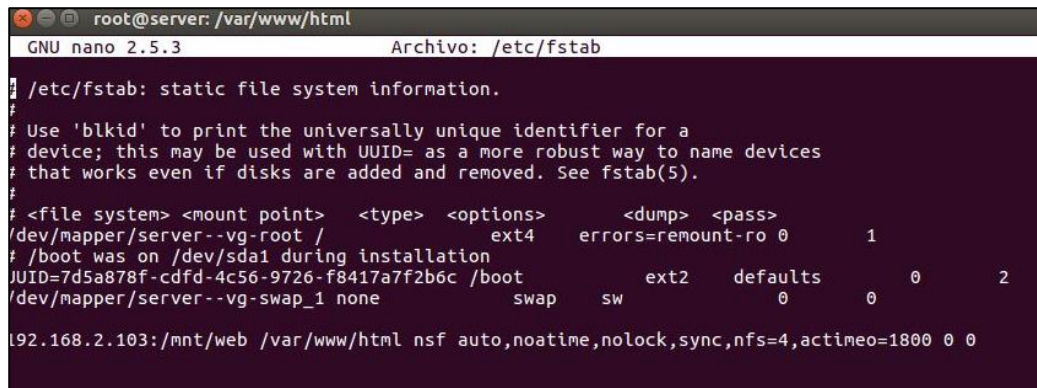
La sintaxis varía dependiendo de la versión NFS que vayamos a utilizar. Pero en el fichero aparecen ejemplos con la sintaxis.

6. En último lugar, para que el volumen se monte de forma automática cada vez que el ordenador se encienda, editamos el fichero `/etc/fstab` (Fig.45) explicado en el glosario de términos. Añadimos la siguiente línea
- ```

192.168.2.103:/mnt/web /var/www/html nfs
auto,noatime,nolock,sync,nfs=4,actimeo=1800 0 0

```





```

root@server: /var/www/html
GNU nano 2.5.3 Archivo: /etc/fstab

/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).
#
<file system> <mount point> <type> <options> <dump> <pass>
/dev/mapper/server--vg-root / ext4 errors=remount-ro 0 1
/boot was on /dev/sda1 during installation
UUID=7d5a878f-cdfd-4c56-9726-f8417a7f2b6c /boot ext2 defaults 0 2
/dev/mapper/server--vg-swap_1 none swap sw 0 0
192.168.2.103:/mnt/web /var/www/html nfs auto,noatime,nolock,async,nfs=4,actimeo=1800 0 0

```

Fig. 45 Fichero fstab

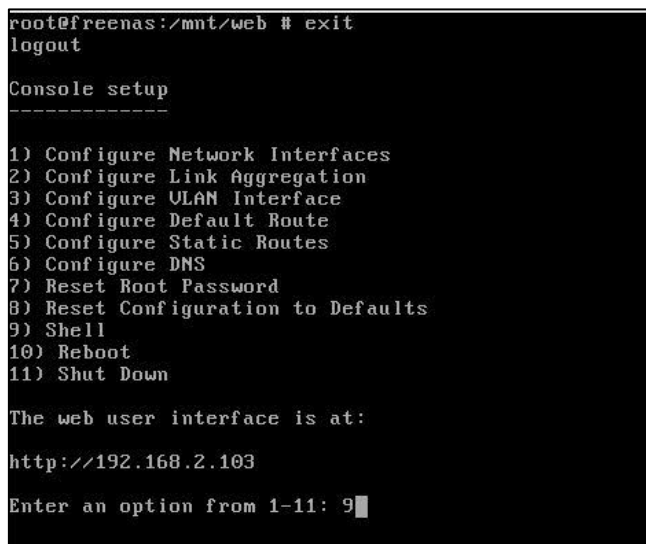
Primero ponemos la IP y el volumen que vamos a montar y después el punto de montaje, seguido de nfs y las opciones que necesitamos poner.

#### Acceso a carpeta compartida

Para comprobar que se ha montado correctamente, crearemos un fichero y una carpeta en el volumen del NAS para ver si se visualiza en el clúster y viceversa.

Primero nos metemos desde el servidor NAS, que nos muestra una serie de opciones.

Elegimos la 9 para abrir una Shell (Fig.46) y poder ejecutar los comandos necesarios:



```

root@freenas:/mnt/web # exit
logout

Console setup

1) Configure Network Interfaces
2) Configure Link Aggregation
3) Configure VLAN Interface
4) Configure Default Route
5) Configure Static Routes
6) Configure DNS
7) Reset Root Password
8) Reset Configuration to Defaults
9) Shell
10) Reboot
11) Shut Down

The web user interface is at:
http://192.168.2.103
Enter an option from 1-11: 9

```

Fig. 46 Interfaz del NAS

Podemos configurar las interfaces de red, la interfaz de VLAN, la ruta por defecto, rutas estáticas, el DNS, resetear la contraseña de root, poner la configuración por defecto, abrir una Shell, reiniciarlo o apagarlo.

Nos situamos en la ruta /mnt/web y ejecutamos los siguientes comandos (Fig.47):

- # `mkdir nas` para crear la carpeta.
- # `touch nas.txt` para crear el fichero.
- Comprobamos que se han creado en el volumen utilizando el comando `ls -ltr`

```
root@freenas:/mnt/web # mkdir nas
root@freenas:/mnt/web # touch nas.txt
root@freenas:/mnt/web # ls -ltr
total 2
drwxr-xr-x 2 root wheel 2 May 26 17:00 jails
drwxr-xr-x 2 root wheel 3 May 26 18:27 prueba
drwxr-xr-x 2 root wheel 2 May 26 18:46 nas
-rw-r--r-- 1 root wheel 0 May 26 18:46 nas.txt
```

Fig. 47 Pruebas

Ahora comprobamos que se ve en la carpeta de los nodos del clúster. Nos situamos en la ruta /var/www/html de ambos nodos (Fig.48 y 49) y ejecutamos `ls -ltr`, como podemos observar el fichero y la carpeta se han creado correctamente.

```
root@server: /var/www/html
root@server:/var/www/html/prueba# cd ..
root@server:/var/www/html# ls -ltr
total 2
drwxr-xr-x 2 root root 2 may 26 17:00 jails
drwxr-xr-x 2 root root 3 may 26 18:27 prueba
drwxr-xr-x 2 root root 2 may 26 18:46 nas
-rw-r--r-- 1 root root 0 may 26 18:46 nas.txt
```

Fig. 48 Comprobación server

```
root@server2: /var/www/html
root@server2:/var/www/html# ls -ltr
total 2
drwxr-xr-x 2 root root 2 may 26 17:00 jails
drwxr-xr-x 2 root root 3 may 26 18:27 prueba
drwxr-xr-x 2 root root 2 may 26 18:46 nas
-rw-r--r-- 1 root root 0 may 26 18:46 nas.txt
```

Fig. 49 Comprobación server2

Tras esto, lo realizaremos en el sentido contrario situándonos en la ruta /var/www/html, desde el nodo1 del clúster (Fig.50).

- # `mkdir nodo1` para crear la carpeta.
- # `touch nodo1.txt` para crear el fichero.
- Por último, ejecutamos el comando `ls -ltr` para comprobar que se han creado.

```
root@server:/var/www/html# mkdir nodo1
root@server:/var/www/html# touch nodo.txt
root@server:/var/www/html# ls -ltr
total 3
drwxr-xr-x 2 root root 2 may 26 17:00 jails
drwxr-xr-x 2 root root 3 may 26 18:27 prueba
drwxr-xr-x 2 root root 2 may 26 18:46 nas
-rw-r--r-- 1 root root 0 may 26 18:46 nas.txt
drwxr-xr-x 2 root root 2 may 26 18:55 nodo1
-rw-r--r-- 1 root root 0 may 26 18:55 nodo.txt
```

Fig. 50 Pureba server

Como hemos hecho anteriormente, nos situamos en la ruta /var/www/html del nodo2 (Fig.51) y comprobamos que existen las carpetas y ficheros que hemos creado.

```
root@server:/var/www/html# mkdir nodo1
root@server:/var/www/html# touch nodo.txt
root@server:/var/www/html# ls -ltr
total 3
drwxr-xr-x 2 root root 2 may 26 17:00 jails
drwxr-xr-x 2 root root 3 may 26 18:27 prueba
drwxr-xr-x 2 root root 2 may 26 18:46 nas
-rw-r--r-- 1 root root 0 may 26 18:46 nas.txt
drwxr-xr-x 2 root root 2 may 26 18:55 nodo1
-rw-r--r-- 1 root root 0 may 26 18:55 nodo.txt
```

Fig. 51 Prueba server

Por otro lado, desde la Shell (Fig.52) abierta anteriormente en el servidor NAS, nos situamos en la ruta del volumen /mnt/web y comprobamos que existen las carpetas y directorios que hemos creado.

```
root@server2:/var/www/html# ls
ejemplo.html jails nas nodo1 nodo.txt prueba
root@server2:/var/www/html#
```

Fig. 52 Prueba server 2

```
root@freenas:/mnt/web # ls -ltr
total 3
drwxr-xr-x 2 root wheel 2 May 26 17:00 jails
drwxr-xr-x 2 root wheel 3 May 26 18:27 prueba
drwxr-xr-x 2 root wheel 2 May 26 18:46 nas
-rw-r--r-- 1 root wheel 0 May 26 18:46 nas.txt
drwxr-xr-x 2 root wheel 2 May 26 18:55 nodo1
-rw-r--r-- 1 root wheel 0 May 26 18:55 nodo.txt
```

Fig. 53 Prueba nas

Una vez hemos realizado esto tendríamos una estructura como la siguiente (Fig.53):

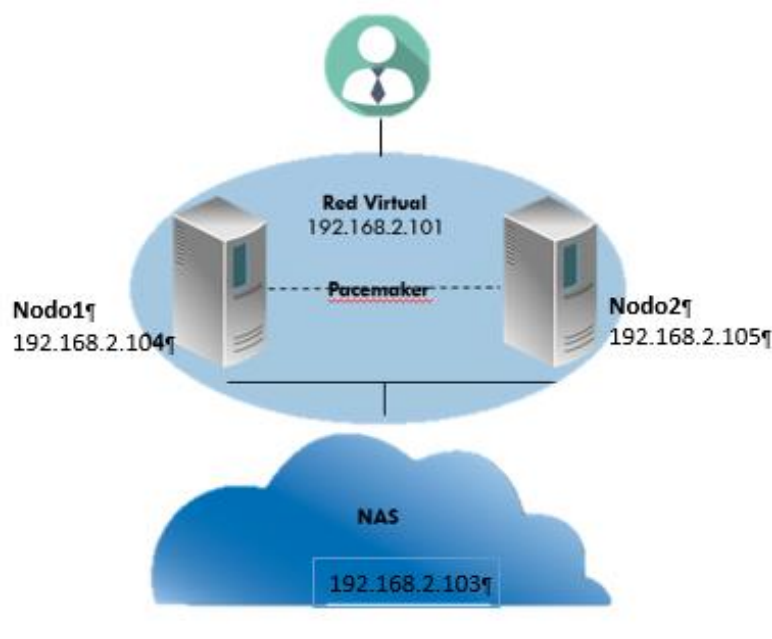


Fig. 54 Estructura del proyecto

## Plan de pruebas:

### 1. Error en Corosync:

Para esto ejecutaremos el comando (Fig.55) `# service corosync stop`

```
server@server:~$ sudo service corosync stop
[sudo] password for server:
server@server:~$
```

**Fig. 55 Para corosync**

Como podemos observar en la (Fig.56) nuestro server se ha apagado ya que el servicio de cluster a “fallado”.

```
Last updated: Mon Jun 4 09:19:07 2018 Last change: Fri Jun 1
2018 by root via cibadmin on server
Stack: corosync
Current DC: server2 (version 1.1.14-70404b0) - partition WITHOUT quorum
2 nodes and 2 resources configured

Online: [server2]
Offline: [server]

Resource Group: mi_cluster_web
 FAILOVER-ADDR (ocf::heartbeat:IPaddr2): Started server2
 FAILOVER-APACHE (lsb:apache2): Started server2
```

**Fig. 56 Estado del clúster**

Como podemos observar el server2 ahora recibir las peticiones, por lo tanto, nuestro servido apache sigue en funcionamiento (Fig.57).



**Fig. 57 Prueba de Apache**

## 2. Fallo en Pacemaker:

Para provocar un fallo en pacemaker lo que haremos será, como en el caso anterior, parar el servicio de pacemaker, así que ejecutaremos el comando

```
service pacemaker stop
```

Y si revisamos el crm con el comando (Fig.58) `# crm_mon`.

```
Last updated: Mon Jun 4 09:22:00 2018 Last change: Fri Jun 1
2018 by root via cibadmin on server
Stack: corosync
Current DC: server2 (version 1.1.14-70404b0) - partition with quorum
2 nodes and 2 resources configured

Online: [server2]
Offline: [server]

Resource Group: mi_cluster_web
 FAILOVER-ADDR (ocf::heartbeat:IPaddr2): Started server2
 FAILOVER-APACHE (lsb:apache2): Started server2
```

*Fig. 58 Estado del clúster*

Y como ocurría en el caso anterior el server2 mantiene nuestro clúster activo y en funcionamiento.

## 3. Fallo en apache.

Como hemos estado haciendo anteriormente para simular este tipo de fallos apagaremos el servicio. `# service apache2 stop` y reiniciaremos tanto corosync como pacemaker ya que al ser algo que el propio sistema realiza por sí mismo deberemos de actualizar su estado. Volvemos a comprobar el estado de nuestro cluster (Fig.59).

```
Last updated: Mon Jun 4 10:33:16 2018 Last change: Fri Jun 1
2018 by root via cibadmin on server
Stack: corosync
Current DC: server (version 1.1.14-70404b0) - partition with quorum
2 nodes and 2 resources configured

Online: [server server2]

Resource Group: mi_cluster_web
 FAILOVER-ADDR (ocf::heartbeat:IPaddr2): Started server
 FAILOVER-APACHE (lsb:apache2): Started server
```

*Fig. 59 Estado del clúster*



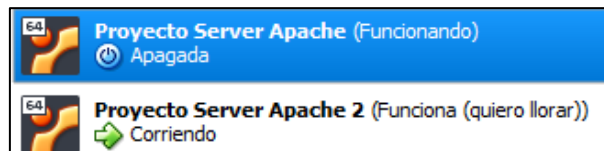
Como podemos observar ahora nuestro apache está iniciado en el server(nodo1).

#### 4. Fallo del servidor:

Para simular esto lo que haremos será apagar un nodo (Fig.61 y 60).



**Fig. 61** Apagado de servidor



**Fig. 60** Apagado de servidor

Como podemos observar en nuestro caso hemos apagado el nodo1 o llamado también server.

Revisamos el crm con el comando (Fig.62) `# crm_mon.`

```
Last updated: Mon Jun 4 09:24:25 2018 Last change: Fri Jun 1
2018 by root via cibadmin on server
Stack: corosync
Current DC: server2 (version 1.1.14-70404b0) - partition WITHOUT quorum
2 nodes and 2 resources configured

Online: [server2]
Offline: [server]

Resource Group: mi_cluster_web
 FAILOVER-ADDR (ocf::heartbeat:IPaddr2): Started server2
 FAILOVER-APACHE (lsb:apache2): Started server2
```

**Fig. 62** Estado del clúster

Como podemos observar nuestro apache y cluster está activo en el server2 por lo que seguimos manteniendo el servicio.

## Conclusiones propuestas

Con la realización de este proyecto hemos aprendido una gran cantidad de conocimientos, que no podrían haber sido asimilados sin la realización de la práctica. Podemos destacar, la creación y configuración de un NAS y de un clúster y sobre todo la importancia de la alta disponibilidad en el entorno empresarial; ya que las empresas cada hora o incluso cada minuto (grandes empresas), generan una gran cantidad de dinero que surge de los servicios que ofrecen, por lo tanto, si alguno de estos falla puede suponer una gran pérdida a la empresa, además, de una mancha en su fiabilidad con respecto a los clientes tanto presentes como futuros.

Por otra parte, el objetivo del proyecto lo hemos cumplido. Hemos creado un clúster de alta disponibilidad con almacenamiento en NAS, capaz de visualizar páginas web gracias al servicio proporcionado de Apache.

Con respecto a los aspectos de mejora, nos gustaría haber podido configurar correctamente el fichero fstab y no hacerlo mediante un script, ya que sería una forma más limpia de hacerlo.

Por último, el resultado final del proyecto ha sido satisfactoria (aunque se podría mejorar), ya que nos ha supuesto la superación de un reto para nosotros debido a que nunca habíamos montado un clúster.

## Bibliografía y referencias

Clúster, pacemaker y corosync:

- <https://www.elarraydejota.com/cluster-apache-en-alta-disponibilidad-con-pacemakercorosync-en-ubuntu-server-lts-14-04/> 08/05/2018
- <https://aula128.wordpress.com/2015/02/28/alta-disponibilidad-como-configurar-un-cluster-ha-linux-con-corosync-y-pacemaker-con-recurso-apache2/> 08/05/2018
- [https://es.wikipedia.org/wiki/Cl%C3%BAster\\_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Cl%C3%BAster_(inform%C3%A1tica)) 22/05/2018

NAS:

- <http://www.freenas.org/> 09/05/2018
- <http://masdefreenas.blogspot.com.es/2015/06/freenas-cuales-son-los-prerequisitos-de.html> 10/05/2018
- <https://www.redeszone.net/2013/02/11/freenas-manual-de-instalacion-y-configuracion/> 10/05/2018
- <https://www.youtube.com/watch?v=2gmbVbTQOB4> 11/05/2018

Configuración NFS

- <https://www.atarea.es/software/utilidades/compartir-directorios-red-local/> 22/05/2018

Otros:

- [https://es.wikipedia.org/wiki/Alta\\_disponibilidad](https://es.wikipedia.org/wiki/Alta_disponibilidad) 22/05/2018
- [https://es.wikipedia.org/wiki/Servidor\\_HTTP\\_Apache](https://es.wikipedia.org/wiki/Servidor_HTTP_Apache) 22/05/2018



## ANEXO I – ERROR FICHERO FSTAB

A la hora de realizar el proyecto nos hemos encontrado con una pequeña dificultad.

Como hemos explicado anteriormente, añadimos el comando mount en fichero fstab para que el volumen del NAS se montara en el punto de montaje del clúster de forma automática cuando el equipo se iniciara.

En un principio lo hicimos de esa forma, hasta que nos dimos cuenta de que no se montaba cuando hicimos la prueba del apagado de un nodo.

Al reiniciar el servidor, entraba en modo consola y aparecía el siguiente error (Fig.63).

```
[OK] Started iSCSI initiator daemon (iscsid).
 Starting Login to default iSCSI targets...
[FAILED] Failed to start NFS Mount Daemon.
See 'systemctl status nfs-mountd.service' for details.
[DEPEND] Dependency failed for NFS server and services.
[OK] Started Login to default iSCSI targets.
[OK] Reached target Remote File Systems (Pre).
[OK] Reached target Remote File Systems.
```

Fig. 63 Erro al iniciar servidor

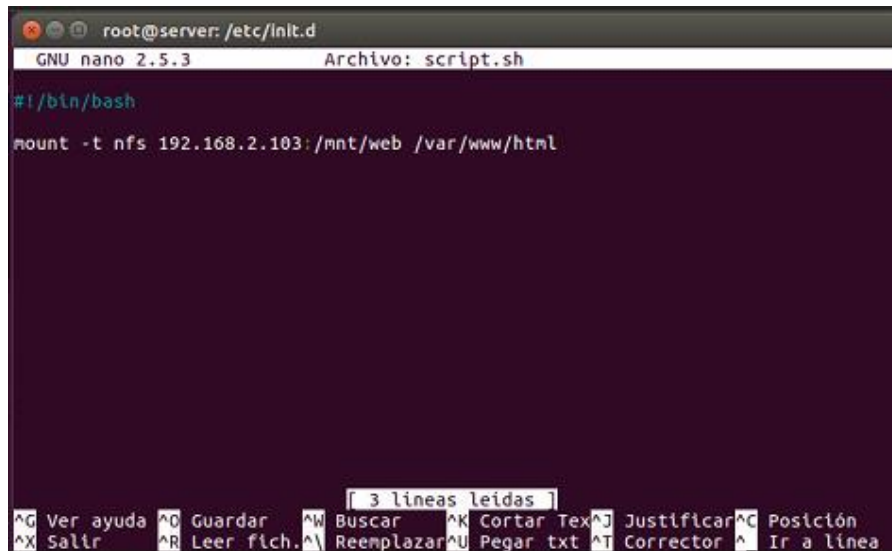
Como teníamos un snapshot del punto anterior, la restauramos y probamos a eliminar la línea de montaje del fichero fstab y creamos un script (Fig.64) para añadirlo posteriormente en /etc/init.d y crear un enlace simbólico al directorio rc3.d

El directorio /etc/init.d es la carpeta donde están los script de arranque y parada de los diferentes servicios o servidores que están instalados en el equipo.

```
root@server2:/home/server2# cd /etc/init.d
root@server2:/etc/init.d# touch script.sh
root@server2:/etc/init.d# nano script.sh
```

Fig. 64 Creación script

Una vez que hayamos creado el fichero del script, procedemos a añadir la sintaxis necesaria para que se monte el volumen del NAS (Fig.65).



```

root@server: /etc/init.d
GNU nano 2.5.3 Archivo: script.sh

#!/bin/bash
mount -t nfs 192.168.2.103:/mnt/web /var/www/html

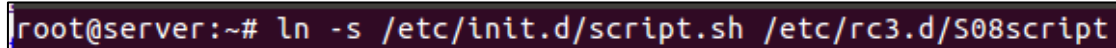
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Tex ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Corrector ^_ Ir a línea

```

Fig. 65 Script

Ponemos el mismo comando que hemos utilizado anteriormente cuando lo hemos montado a mano. Una vez que lo tengamos, guardamos y cerramos.

Tras esto creamos un enlace simbólico (Fig.66) en el runlevel (explicado en el glosario de términos) correspondiente, en nuestro caso rc3.d, que es el nivel de multiusuario con soporte de red.



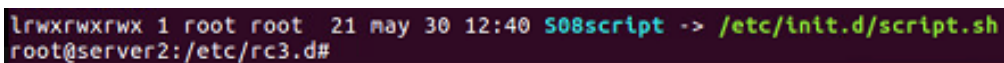
```

root@server:~# ln -s /etc/init.d/script.sh /etc/rc3.d/S08script

```

Fig. 66 Enlace simbólico

Cuando este creado debería quedar como en la imagen siguiente (Fig.67).



```

lrwxrwxrwx 1 root root 21 may 30 12:40 S08script -> /etc/init.d/script.sh
root@server2:/etc/rc3.d#

```

Fig. 67 Enlace simbólico