

**UNIVERSIDAD LUTERANA SALVADOREÑA
FACULTAD DE CIENCIAS DEL HOMBRE Y LA NATURALEZA
LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN**



MATERIA:

REDES II

TEMA DEL PROYECTO

IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA DISPONIBILIDAD

CATEDRATICO:

INGENIERO MANUEL FLORES VILLATORO

ESTUDIANTES:

NOMBRES	CARNET	PARTICIPACION
PEDRO ANTONIO TREJO NOBLE	TN01121295	100 %
NICOLAS GREGORIO CABRERA	GC01121355	100%

SAN SALVADOR, 21 DE NOVIEMBRE DE 2015

INDICE

Contenido

1 INTRODUCCION	4
2 OBJETIVOS.....	6
a) Objetivo General.....	6
b) Objetivos específicos	6
3 MARCO TEORICO	6
3.1 El concepto de alta disponibilidad (HA).....	6
3.3 Alta Disponibilidad en las organizaciones.....	7
3.4 Determinación de las necesidades de disponibilidad del cliente.	7
3.5 Niveles de disponibilidad	9
3.6 Causas posibles del tiempo de inactividad o interrupción.....	10
3.7 Interrupciones no planificadas	11
3.8 Interrupciones planificadas.....	11
3.9 ¿Cómo se puede implementar alta disponibilidad?	12
3.10 Sistema de alta disponibilidad y sistema tolerante a fallos.....	12
4 TECNOLOGIAS UTILIZADAS PARA REALIZAR ESTE PROYECTO.....	13
4.1 Sistema operativo	13
4.2 Distribución de datos.....	14
4.3 Replicación de archivos	15
4.5 Lsyncd	15
4.5.1 El algoritmo LSYNCD	16
4.6 Gestión de monitorización.....	16
4.6.1 Heartbeat	17
4.6.2 Estos 3 ficheros son:.....	18
4.7 Vsftpd.....	20
4.7.1 Protocolos utilizados por vsftpd.....	22
4.7.2 Protocolo SSL, para la seguridad de dichas transferencias.....	22
5 TERMINOLOGIA.....	23
6. INFORMACIÓN SOBRE LA CONSTRUCCIÓN DEL PROYECTO, DETALLANDO CADA UNO DE LOS PASOS, COMANDOS, ARTEFACTOS, PROCEDIMIENTOS QUE SE REALIZARON.....	24
6.1 Instalación de Heartbeat.....	24
6.2 Instalación de vsftpd.....	34

6.3 Instalación de Incron	36
6.4 Sincronización de archivos.....	37
7 CONCLUSIONES	41
8 BIBLIOGRAFIA.....	42
ANEXOS	43

1 INTRODUCCION

Con el actual ritmo de crecimiento del comercio y el movimiento de datos de todo tipo en Internet (más de un 100% anual) y la incuestionable importancia de la informática en las empresas actuales de cualquier tamaño, es cada día más importante que los sistemas informáticos de éstas puedan funcionar de forma ininterrumpida y sin errores las 24h del día, 7 días a la semana y 365 días al año, ya sea para dar soporte interno (contabilidad, control de personal, desarrollo...) como para ofrecer servicios a través de Internet (comercio electrónico, correo, portales, etc). A esta necesidad de un servicio ininterrumpido y fiable se le conoce como alta disponibilidad, que es de lo que trata nuestro proyecto.

En la actualidad los sistemas informáticos de las entidades públicas y privadas son una pieza imprescindible para su correcto funcionamiento. Cada vez más empresas y organismos públicos confían procesos de negocio críticos en sistemas informáticos para mejorar la productividad y disponer de esta información crítica en un tiempo mínimo.

Confiar los procesos de negocio en los sistemas informáticos de la organización aporta muchas ventajas pero introduce nuevos puntos de fallo que necesitan estar controlados.

Dos estudios independientes realizados en 1995 por Oracle Corp. y Datamation revelaron que una empresa media pierde entre 80,000 y 350,000 dólares por hora de interrupción no planeada de sus servicios informáticos. Otro ejemplo de la necesidad de la alta disponibilidad es que tras el primer atentado en el World Trade Center el 26 de febrero de 1993, 145 de las 350 empresas que allí se hospedaban (algo más del 40%) tuvieron que cerrar sus puertas tras este incidente por no disponer de una infraestructura informática redundante.

La principal técnica para obtener estos sistemas tolerantes a fallos es la redundancia, estrategia utilizada en la industria aeronáutica prácticamente desde sus principios, que consiste en replicar las zonas críticas del sistema, teniendo una unidad activa y varias copias inactivas que, tras el fallo de la principal, sean capaces de retomar su labor en el punto que aquella falló, en el menor tiempo posible y de forma transparente para el usuario.

Para evitar fallos de servicio en los sistemas informáticos de las organizaciones aparecen los sistemas de alta disponibilidad (HA). Un sistema de alta disponibilidad (HA) está formado por distintos componentes hardware y software que combinados proporcionan acceso de forma ininterrumpida a los procesos de negocio. En la actualidad, el uso de sistemas de alta disponibilidad está muy extendido en las organizaciones y esto nos motiva y justifica el presente trabajo.

2 OBJETIVOS.

a) Objetivo General

Configurar un servicio de alta disponibilidad (HA) utilizando Heartbeat, para que exista un servidor de fallos que sirva de respaldo en caso de que el servidor principal deje de funcionar.

b) Objetivos específicos

- ✓ Comprender el funcionamiento de Heartbeat mediante la configuración de sus servicios.
- ✓ Utilizar el sistema operativo Debian versión 8.0 para la instalación y configuración del Servicio de Alta Disponibilidad Heartbeat.
- ✓ Utilizar todas las técnicas necesarias para asegurar la estabilidad de cada uno de los servidores del clúster.
- ✓ Implantar en el clúster un sistema real con el fin de detectar y conocer los diferentes problemas que ocurren en este tipo de proyectos para buscarles una solución.

3 MARCO TEORICO

3.1 El concepto de alta disponibilidad (HA)

A continuación se va a realizar una descripción del concepto de alta disponibilidad para facilitar la comprensión de los siguientes apartados del proyecto en los que se diseñará e implementará una solución de clúster de alta disponibilidad para una aplicación empresarial.

Se comenzará definiendo el concepto de alta disponibilidad y el motivo por el que es una herramienta imprescindible para multitud de organizaciones.

A continuación, se estudiará la importancia de la determinación de las necesidades de disponibilidad reales del cliente para abordar un proyecto de alta disponibilidad lo más coherente posible.

En el siguiente apartado se describirán los distintos niveles de disponibilidad empleando un punto de vista general y un punto de vista cercano al usuario.

El último apartado del presente se describe aspectos importantes para realizar

una correcta planificación para asegurar un nivel determinado de disponibilidad en un sistema.

3.3 Alta Disponibilidad en las organizaciones

Un sistema se encuentra disponible si los usuarios pueden realizar operaciones sobre éste (acceder al sistema, someter nuevos trabajos, actualizar o modificar trabajos existentes,...).

En la actualidad los departamentos de tecnología de la información de las organizaciones han ganado mayor protagonismo y operaciones de negocio críticas que antes no dependían de estos departamentos ahora lo hacen. Debido a esta tendencia, es necesario asegurar un servicio continuado y de calidad a los procesos de negocio críticos que se ejecutan en los sistemas informáticos de las organizaciones.

El concepto de alta disponibilidad comprende un diseño de sistema y la implementación de protocolos asociados que aseguren un cierto nivel de continuidad operacional durante un periodo de tiempo determinado.

La disponibilidad de un sistema suele estar medida por año y únicamente tiene en cuenta los cortes de servicio no planificados. No se contemplan las paradas planificadas de los sistemas para el cálculo de la disponibilidad. En niveles muy altos de disponibilidad se pueden introducir estos dos parámetros (paradas planificadas y no planificadas) para el cálculo del valor.

3.4 Determinación de las necesidades de disponibilidad del cliente.

El principal aspecto en el diseño de la disponibilidad de un sistema es obtener los requerimientos reales de disponibilidad de los usuarios. Para obtener esta información es necesario estudiar con detenimiento el modo en el que los usuarios utilizan las aplicaciones de la organización, determinar cuáles son críticas y necesitan de una mayor disponibilidad.

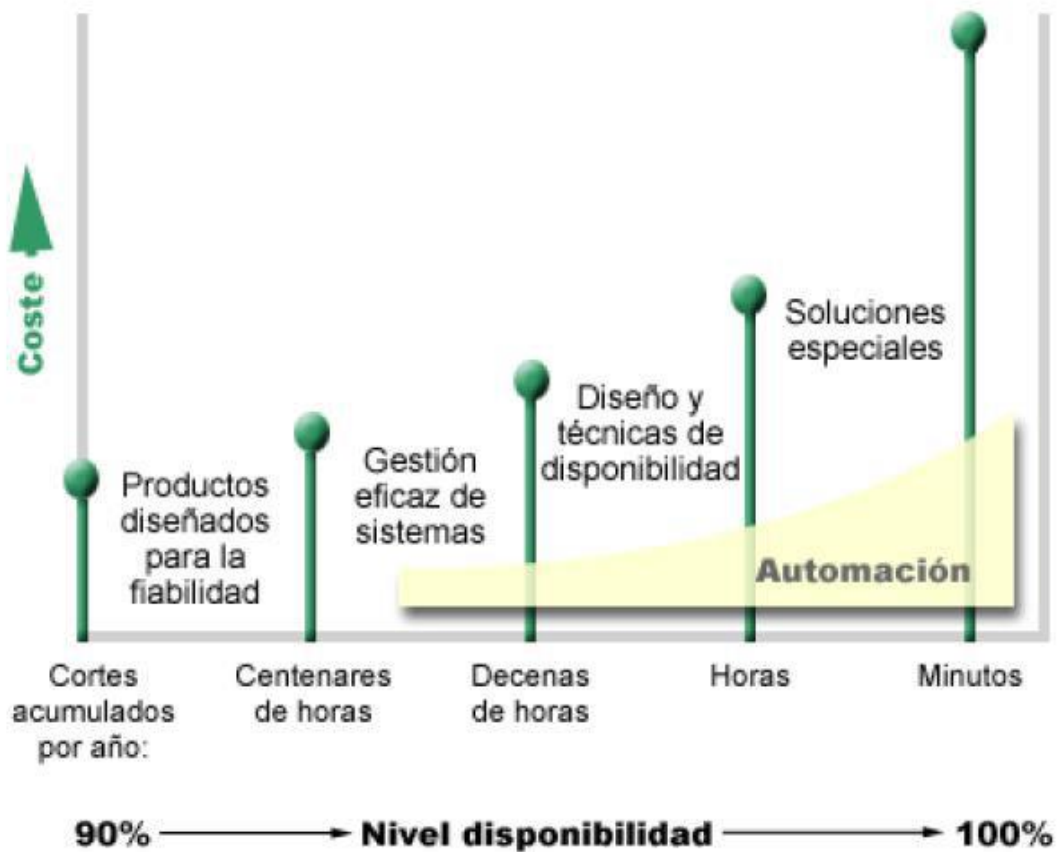
En este primer estudio del entorno es recomendable que el cliente sea consciente del coste de hacer que un sistema junto con sus aplicaciones sea altamente disponible, ya que como norma general, el coste de un sistema se incrementa exponencialmente conforme aumenta su disponibilidad.

Tras realizar el estudio de las necesidades de disponibilidad del cliente, se

puede redactar un documento de acuerdo de nivel de servicio. En esta primera fase de estudio el documento únicamente debe reflejar los aspectos relativos a la disponibilidad de los sistemas. Conforme se va avanzando en el diseño de la solución, el documento de acuerdo de nivel de servicio reflejará otros aspectos como el tiempo de respuesta tras un fallo en el sistema, las paradas planificadas necesarias para el correcto mantenimiento del sistema y otros aspectos relativos al rendimiento o calidad del servicio.

En la mayoría de los casos, puede obtenerse un alto nivel de disponibilidad implementando los procesos y los métodos de gestión de sistemas adecuados. Cuanta más disponibilidad continuada sea necesaria, mayor será la inversión a realizar. Antes de realizar una inversión de este tipo, es necesario asegurarse que ese nivel de disponibilidad es realmente necesario. En la figura siguiente se muestra cómo distintas técnicas pueden mejorar la disponibilidad, aunque también pueden aumentar el precio que se deberá pagar..

En la siguiente figura vemos el coste de diferentes técnicas para mejorar la disponibilidad:



3.5 Niveles de disponibilidad

Existen varios niveles de disponibilidad. Estos niveles se diferencian por el tipo y la duración de las interrupciones que admiten. Estos niveles son los siguientes:

- ✓ **Altamente disponible.** El servidor ofrece un nivel aceptable o acordado de servicio durante su período de funcionamiento programado. El objetivo es que el servidor esté disponible cuando el cliente lo necesite.
- ✓ **Alta disponibilidad.** El servidor ofrece un nivel aceptable o acordado de servicio durante su período de funcionamiento programado. El objetivo es que no se produzca ninguna interrupción no planificada; podrían producirse algunas interrupciones planificadas.

- ✓ **Operaciones continuadas.** El servidor ofrece un nivel aceptable o acordado de servicio 24 horas al día, 365 días al año. El objetivo es que el servidor funcione sin interrupciones planificadas; podrían producirse algunas interrupciones no planificadas.

En el diagrama siguiente se muestra cómo se relacionan entre sí estos distintos niveles de disponibilidad y qué tipos de empresas son adecuados para cada nivel.



3.6 Causas posibles del tiempo de inactividad o interrupción

En los escenarios más críticos de disponibilidad no se puede permitir que el mantenimiento planificado y los tiempos de parada no planificados interrumpen la disponibilidad del sistema en cualquier momento. A continuación se

examinan cuáles son esas posibles causas, tanto imprevistas como previstas que se deben tener en cuenta al diseñar una solución altamente disponible.

3.7 Interrupciones no planificadas

En cuanto a fallos imprevistos podemos considerar fallos de software, fallos de hardware, errores humanos, y desastres naturales.

- ✓ Los fallos del software incluyen sistema operativo, base de datos, middleware, uso y fallos de la red. Un fallo de estos componentes puede causar un fallo del sistema.
- ✓ Los fallos del hardware incluyen errores de sistema (CPU, memoria, suministro eléctrico, bus), periféricos (disco, cinta, controladoras), fallos en la red, y apagones.
- ✓ Errores humanos, causa principal de fallos, incluyen errores de operador, de usuario, de administrador de la base de datos, o de administrador del sistema. Otro tipo de errores humanos que puede causar el tiempo de inactividad no planificado son los de sabotaje.
- ✓ La categoría final es desastres naturales. Aunque infrecuentes, estas causas del tiempo de inactividad pueden tener impactos extremos en las empresas, debido a su efecto prolongado sobre operaciones. Las causas posibles de desastres incluyen fuegos, inundaciones, terremotos, apagones, y bombardeos.

3.8 Interrupciones planificadas

Las interrupciones planificadas son necesarias y se cuenta con ellas; sin embargo, el hecho de que sean planificadas no significa que no sean disruptivas. Las interrupciones planificadas suelen estar relacionadas con el mantenimiento del sistema. En estos casos es importante diseñar un sistema para reducir al mínimo las interrupciones. Las causas previstas de tiempo de

inactividad incluyen operaciones rutinarias, mantenimiento periódico y nuevos despliegues.

- ✓ Operaciones rutinarias como la instalación de parches o reconfigurar el sistema son ocasionalmente necesarias para actualizar la base de datos, aplicaciones, SO, middleware o la red.
- ✓ Mantenimiento periódico de la Base de Datos (del sistema de almacenamiento, parámetros de inicialización, parches de software), de las aplicaciones (administración del esquema, parches de software), del SO, Middleware, de la red.
- ✓ Despliegues nuevos significan actualizaciones hardware, SO, base de datos, Middleware, de aplicaciones y de red.

También es importante considerar no sólo el tiempo utilizado en realizar la actualización sino el efecto que el cambio repercute en la aplicación en general.

3.9 ¿Cómo se puede implementar alta disponibilidad?

A continuación se señalan algunas de las técnicas utilizadas para conseguir un sistema con la más alta disponibilidad. La principal técnica para obtener estos sistemas se centra en la redundancia y en replicar las zonas críticas, consiguiendo una unidad activa y varias copias inactivas que tras el fallo de la principal sean capaces de retomar su labor en el punto que aquella falló en el menor tiempo posible y de forma transparente para el usuario.

3.10 Sistema de alta disponibilidad y sistema tolerante a fallos

En un sistema tolerante a fallos, cuando se produce un fallo en hardware, el hardware asociado a este tipo de sistema es capaz de detectar el subsistema que falla y obrar en consecuencia para restablecer el servicio en segundos (o incluso decimas de segundos). El cliente del servicio no notará ningún tiempo de fuera de servicio.

En los sistemas de alta disponibilidad existen los tiempos de fuera de servicio, son mínimos pero existen, van desde 1 minuto hasta 5 o 10 minutos, según sea el caso. En teoría esta es la única diferencia entre ambos. Pero en los últimos años, se ha ido acercando la idea de alta disponibilidad a la idea de tolerancia a fallos, debido al abaratamiento del hardware, y de ciertas tecnologías que han ido surgiendo.

En la mayoría de los análisis que se hacen de un sistema de servicio, si la aplicación puede estar un mínimo tiempo fuera de servicio, y podemos permitir que el cliente pierda la sesión o la conexión, temporalmente la alta disponibilidad es una opción muy apropiada. Hay soluciones de alta disponibilidad en las cuales las conexiones se mantienen y las sesiones se recuperan.

4 TECNOLOGIAS UTILIZADAS PARA REALIZAR ESTE PROYECTO

4.1 Sistema operativo

Los dos servidores van a utilizar LINUX como sistema operativo, más concretamente Debian 8.0. Hemos elegido este sistema operativo por su seguridad y disponibilidad de buen software para nuestro proyecto y además por su licencia GPL, Debian es considerado como muy seguro en el aspecto de servidores.

¿Por qué elegir Linux para la implementación de un Clúster?

Debian GNU/Linux: Es un sistema operativo gratuito, una de las distribuciones de Linux más populares e influyentes. Es conocido por su adhesión a las filosofías del software libre y por su abundancia de opciones (su actual versión incluye más de 43 mil paquetes de software). Debian GNU/Linux, también es base para otras múltiples distribuciones de Linux como Knoppix, Linspire, MEPIS, Xandros y la familia Ubuntu. Debian también es conocido por su sistema de gestión de paquetes (especialmente APT), por sus estrictas políticas con respecto a sus paquetes y la calidad de sus lanzamientos. Estas prácticas permiten fáciles actualizaciones entre lanzamientos, y una instalación y remoción sencilla de paquetes.

La evolución y estabilidad que ha alcanzado el SO Linux, es muy importante al desarrollo de muchas tecnologías nuevas, entre ellas la de Clúster.

Linux es conocida, como una plataforma de computación estable y supercomputación muy consolidada.

Linux posee una de las pilas TCP/IP más completas y estables que existen actualmente. La característica de IP Aliasing de Linux, permite asignar varias direcciones IP a una misma interfaz, esto nos permite poder levantar una dirección IP en un nodo del cluster. Con estos servicios el cluster puede ser consciente de la topología de la red y reaccionar ante las caídas de líneas de comunicaciones o nodos.

Adicionalmente, la naturaleza de fuente abierta del sistema Linux ha permitido a los programadores, añadir directamente características adicionales al sistema operativo para responder a las necesidades de los cluster de computación. Dándonos por entender que Linux es un sistema evolutivo y ajustable.

Linux se preocupa para no dejar obsoleto el hardware que no es de última tecnología.

Es de destacar el enorme esfuerzo que está realizando la comunidad Linux mundial en la difusión de documentación, manuales, tutoriales paso a paso para principiantes o textos especializados para expertos en la materia.

4.2 Distribución de datos

La alta disponibilidad implica que los datos que tengan que servir o procesar deben estar disponibles para todos y cada uno de los servidores, de forma que para el usuario el cluster se comporte como un único ordenador, en el que ellos copian en un único lugar los ficheros, y el software de control del cluster internamente se encarga de hacer llegar una copia a cada uno de los servidores que lo componen.

Para esta acción existen dos estrategias: La replicación física de archivos, en la

que cada servidor tendrá una copia de todos los datos en su disco duro; y la distribución de los datos mediante sistemas de archivos distribuidos. En el que un servidor de ficheros y el resto de equipos del cluster accederán a sus contenidos por la red.

4.3 Replicación de archivos

Esta es la alternativa más “primitiva” para la distribución del contenido a servir a todos los equipos del cluster, es la replicación (automática o manual) de los ficheros en todos los ordenadores.

En nuestro proyecto aplicamos un novedoso protocolo que optimizará en gran medida la cantidad de datos a transmitir por la red y, en consecuencia, el tiempo necesario para realizar la sincronización.

4.5 Lsyncd

Es un programa para la sincronización de archivos de forma eficiente, transfiriendo solamente las partes que son diferentes (que pueden ser un porcentaje muy pequeño) y además las puede comprimir (-z) LSYNCD puede trabajar como cliente y como servidor.

Lsyncd es una implementación como servicio de rsync que permite monitorear un sistema de archivos (por ejemplo el directorio web) para mantenerlo sincronizado entre varios servidores y ofrecer siempre el mismo contenido actualizado. Muy útil para una configuración con tolerancia a fallos y/o balanceo de carga.

Con LSYNCD podemos:

- ✓ Copiar o sincronizar ficheros, directorios o sistemas de archivos enteros, manteniendo enlaces, permisos, fechas, etc.
- ✓ Redireccionar todo el tráfico a través de ssh para cifrarlos.
- ✓ Permitir un acceso “anónimo” para que terceras personas puedan hacer mirror de nuestras páginas.

- ✓ Copiar por la red únicamente las diferencias entre los ficheros.

4.5.1 El algoritmo LSYNCD

El algoritmo LSYNCD funciona a grandes rasgos de la siguiente forma:

El equipo receptor ('A') divide el fichero en bloques de un tamaño fijo que no se solapan entre si.

Se dispone de dos algoritmos de cálculo CRC, uno muy rápido ('X') pero no exacto, aunque asegura que nunca dará un falso negativo (un bloque con CRC correcto siempre será evaluado como correcto; uno incorrecto pueda que sea evaluado como correcto).

Además, este algoritmo tiene la característica de que el resultado del bloque X+1 se puede calcular rápidamente a partir del resultado del bloque X.

Otro más lento ('Y'), pero que sí que es capaz de discriminar siempre si el CRC es correcto o no. Se calculan los valores de ambos algoritmos sobre los bloques del fichero, y se envían al otro equipo.

El equipo emisor ('B') busca en su fichero bloques del tamaño fijado para los que coincida el algoritmo 'X'; si el CRC difiere en el fichero local y el remoto, no hace falta retransmitir ese bloque, si coincide, se analiza con el algoritmo 'Y'.

Si con el algoritmo 'Y' también coincide, entonces el bloque ha cambiado y habrá que transmitirlo. Se envía al otro equipo la información precisa para reconstruir el fichero, bien como una referencia a otro bloque del fichero o comodatos puros.

De esta forma, se consigue disminuir en gran medida la cantidad de datos a transmitir entre las dos máquinas, algo muy a tener en cuenta si la conexión entre los equipos es lenta o si el número de ficheros asincronizar es grande.

4.6 Gestión de monitorización

La monitorización de servicios es un aspecto muy importante en el clustering de alta disponibilidad: ya que si algún servidor falla, es fundamental advertirlo

de alguna forma y efectuar las acciones pertinentes (eliminarlo de la lista de servidores activos y hacer que algún otro servidor tome el lugar de este).

4.6.1 Heartbeat



Traducido del inglés significa Latido de corazón

Heartbeat se utiliza para monitorizar y significa “latido de corazón”, lo que hace es periódicamente enviar un paquete, que si no llegara indicaría que un servidor no está disponible por lo tanto se sabe que el servidor ha caído y se toman las medidas necesarias.

Cuando un servidor deja de hacer heartbeats y se considera muerto, se hace una transición en el cluster. La mayoría de los mensajes de manejo del cluster que no son heartbeats se realizan durante estas transiciones.

Los mensajes de heartbeat se envían por todas las líneas de comunicación a la vez, así si una línea de apoyo cae, se avisará de ese problema antes de que la línea principal caiga y no haya una línea secundaria para continuar el servicio.

Heartbeat tiene el problema que si no tiene una línea dedicada, aunque esta sea una línea serie al tener un tráfico que aunque pequeño, es constante, suele dar muchas colisiones con otros tráficos que puedan ir por la misma red.

Heartbeat proporciona un mecanismo para que dos servidores controlen su estado mutuamente, a través de varios accesos: red Ethernet, cable serie, etc.

De esta forma, al tener varias conexiones y realizar la comprobación por todas ellas no se incurrirán en errores como creer que un servidor ha caído cuando realmente lo que existe es un problema con la red.

La comprobación de estado se realiza a nivel de aplicación, ya que los servicios heartbeat en cada máquina se comunican entre sí mediante un protocolo propio, que además va cifrado para asegurar la identidad de cada máquina.

Heartbeat también puede controlar “cuelgues” en la propia máquina, programando algún dispositivo watchdog (hardware o software en el kernel) para reiniciar la máquina de forma automática. El funcionamiento del watchdog es el siguiente: este dispositivo se programa para que tenga que recibir una entrada (algún texto, lo que sea) cada x tiempo. En caso de fallar esta entrada varias veces seguida, el watchdog se encarga de reiniciar la máquina en forma automática.

Heartbeat se puede programar para que él mismo se encargue de conectar con el watchdog con la frecuencia programada para que no reinicie la máquina, y para que deje de hacerlo si detecta algún problema grave en el equipo (o si se cuelga, con lo que dejaría de enviar estas señales y el watchdog reiniciaría la máquina).

Otra característica importante de heartbeat es que puede adueñarse de la IP de otra máquina (la que está monitorizando) mediante la técnica conocida como ARP IP spoofing: cuando detecta que la otra máquina ha caído, comienza a enviar tramas ARP anunciando sus direcciones IP y MAC, con lo que el resto de equipos routers, y demás dispositivos de red asociarán a partir de ese momento la IP indicada con la dirección MAC de la tarjeta del servidor.

La carpeta /heartbeat, contiene 3 ficheros que debemos configurar para que Heartbeat funcione.

4.6.2 Estos 3 ficheros son:

- ✓ **authkeys:** Fichero de autenticación entre nodos.
- ✓ **ha.cf:** Fichero principal de configuración de Heartbeat

- ✓ **haresources:** Fichero donde se indica cuál será el nodo activo, así como los servicios a gestionar.

Debemos descomprimirlos y copiarlos a la carpeta /etc/ha.d para su configuración. Comenzaremos con la configuración de los ficheros.

Authkeys: En este fichero estableceremos la contraseña de autenticación compartida por los dos nodos. Para generarla hemos utilizado la herramienta cryptool.

Le daremos permisos para que solo pueda ser accedido por el propietario del mismo.

ha.cf: Este es el fichero principal de configuración de Heartbeat. Aquí descomentaremos algunas líneas que son esenciales para el correcto funcionamiento de Heartbeat.

Tiempo transcurrido entre el envío de cada uno de los latidos (segundos)
keepalive 2.

Tiempo transcurrido hasta declarar al nodo como caído (segundos) deadtime
30.

Tiempo transcurrido para comenzar a levantar los servicios (segundos)
initdead 120.

Puerto para la comunicación UDP udpport 694.

Interfaz utilizado para enviar los mensajes de broadcast bcast eth0

#mcast se deja igual, solo se descomenta.

Interfaz y nodo al que enviar los latidos ucast eth0 ha2

Si el nodo cae y se recupera, vuelve a su rol original.

#Servidor principal y secundario.

Haresources: Este fichero contiene información de los recursos que deseamos que tengan alta disponibilidad. Al final de este archivo estableceremos quien será el servidor que predominará, la ip virtual y el servicio que se brindará.

Luego reiniciaremos el servicio Heartbeat con el comando:

```
# service heartbeat restart.
```

4.7 Vsftpd

Es un servidor para sistemas Unix, incluyendo Linux. La licencia de vsftpd la tiene GPL aunque es totalmente gratuito. Vsftpd es un servidor seguro, extremadamente rápido y estable, nosotros podremos ver más adelante en el desarrollo de nuestro proyecto claras evidencias de estas tres afirmaciones, también existen muchas empresas importantes que están satisfechas con el uso de este servidor seguro de FTP.

A pesar de ser pequeño para tener el propósito de velocidad y de seguridad, tiene muchos mecanismos complicados de FTP. Los siguientes mecanismos no es de ninguna manera una lista exclusiva, vsftpd maneja lo siguiente:

- ✓ Configuración de IP virtual
- ✓ Usuarios Virtuales
- ✓ Operación xinetd o funcionamiento Standalone (autónomo)
- ✓ Gran variedad de configuración del usuario
- ✓ Ancho de banda limitado
- ✓ Configuración de fuentes IP
- ✓ Límites de fuentes IP
- ✓ Soporta IPv6
- ✓ Encriptación soportada a través de la integración SSL (Security Sockets Layer, capa de conexión segura)

Vsftpd fue diseñado e implementado desde un principio pensando en la Seguridad. Para ello realiza cosas como:

- ❖ Arregla el diseño defectuoso presente en muchas instalaciones de wu-ftp, pro-ftp y bsd-ftp debido a que no permite el acceso al peligroso usuario de la raíz (root).

- ❖ Hace uso de instalaciones seguras y potentes como las capacidades y chroot.
- ❖ Emplea técnicas de codificación segura solucionando el problema del desbordamiento de buffer.

Si nuestro principal requisito para un servidor FTP es uno de los siguientes:

- ✓ Seguridad.
- ✓ Rápido funcionamiento.
- ✓ Estable.

Entonces Vsftpd es probablemente el servidor FTP que más nos conviene:

Presentación del diseño seguro de vsftpd el cual usa las ventajas de Linux con buen efecto. Las decisiones de diseño tomadas son por ejemplo las siguientes:

1º Todo análisis y actuación sobre datos potencialmente maliciosos de una red remota es hecho un proceso ejecutado como un cliente sin privilegios. Además, este proceso se ejecuta como un chroot () encerrado, asegurando el área de archivos ftp que es accesible por éste.

2º Cualquier operación privilegiada es controlada por un proceso padre privilegiado. El código para este proceso padre privilegiado es lo más pequeño posible por motivo de seguridad y se encarga de que el proceso hijo al cual se le está otorgando privilegios no haga uso malintencionado de este.

3º Este mismo proceso padre privilegiado recibe peticiones de un proceso hijo no privilegiado sobre un socket.

4º Este mismo proceso padre privilegiado hace uso de las capabilities y chroot(), para ejecutar con el mínimo privilegio requerido. Después del login, dependiendo de que opciones se hayan seleccionado, el proceso padre privilegiado dinámicamente calcula que privilegios es requerido. En algunos casos, aunque no tenga importancia el hecho de no tener privilegios.

5º vsftpd-2.x.x soporta SSL y TLS usando OpenSSL. Todo análisis de protocolo

OpenSSL está funcionando en un chroot() encerrado, ejecutándose bajo un cliente sin privilegios, esto es actualmente muy complicado de hacer, pero vsftpd arregla eso en el nombre, siendo por tanto muy seguro.

4.7.1 Protocolos utilizados por vsftpd

Protocolo FTP, para la transferencia de datos.

El servidor vsftpd se comunica con el cliente a través del protocolo FTP. El sistema FTP, que se define para TCP/IP, se basa sobre el servicio de conexiones extremo fiable TCP. Este sistema ha generado un gran volumen de tráfico de información en la red durante los últimos años y aunque en principio no estaba diseñado para usuarios, permite un acceso interactivo a través de la invocación mediante programas. FTP permite la autenticación además de la posibilidad de especificar el formato de los ficheros.

Veamos el modelo de trabajo del proceso FTP. El servidor FTP permite el acceso concurrente, es decir, varias conexiones simultáneas mediante TCP. En el servidor tendremos un proceso, denominado maestro, que está esperando conexiones por parte de los distintos clientes. Cada vez que se conecta un cliente, el proceso maestro crea un proceso esclavo que se encargará de la interacción con el cliente. Este proceso esclavo va a manejar una conexión de control, que permanece durante toda la transferencia de ficheros, y por otro lado una conexión de transferencia de datos, que se crea y destruye cada vez que se transfiere un fichero.

En función de la implementación del servidor es posible que el proceso esclavo cree un proceso específico para gestionar la conexión de datos. En el cliente tenemos un proceso que gestiona la conexión de control y otro proceso, o no, para la conexión de datos. Siempre tenemos que tener 2 conexiones, una fija y otra dinámica.

4.7.2 Protocolo SSL, para la seguridad de dichas transferencias.

El servidor vsftpd proporciona mecanismos de seguridad mediante el uso del protocolo SSL (Secure Sockets Layer), basado en una combinación de técnicas de encriptación y la utilización de un Certificado Digital.

✓ Encriptación

El proceso de encriptación de la información depende del uso de claves, diferenciado según el tipo de criptografía empleada:

Convencional o simétrica: los dos finales de una transacción tienen la misma clave, la cual es usada para decodificar cada una de las otras transmisiones.

Pública o asimétrica: coexisten dos claves, una clave pública y otra clave privada. Los datos codificados con la clave pública sólo pueden ser decodificados con la clave privada y viceversa.

5 TERMINOLOGIA

Clúster: Conjunto de ordenadores construidas mediante la utilización de hardware común y que se comportan como si fuesen una única computadora.

Clúster de alta disponibilidad: Clúster formado por dos o más máquinas que comparten una serie de servicios y están constantemente monitorizándose entre sí.

Nodo: Cada uno de los ordenadores que forman parte del clúster.

Máquina virtual: Software que emula un ordenador y puede ejecutar programas como si fuese un ordenador real.

La nube: Metodología que permite ofrecer servicios a través de internet.

Vsftpd: Es un servidor muy seguro, extremadamente rápido y estable

Ftp: File Transfer Protocol, Protocolo de Transferencia de Archivos

Es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol), basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

Lsync: Es un programa para la sincronización de archivos de forma eficiente,

Heartbeat: Representa un servidor para sistemas operativos y es muy seguro, funciona en plataformas como Linux, BSD, Solaris, Hp-ux-ux e IRIX. Tiene muchas características que faltan en otros servidores FTPs.

6. INFORMACIÓN SOBRE LA CONSTRUCCIÓN DEL PROYECTO, DETALLANDO CADA UNO DE LOS PASOS, COMANDOS, ARTEFACTOS, PROCEDIMIENTOS QUE SE REALIZARON.

6.1 Instalación de Heartbeat

```
Archivo Editar Ver Terminal Pestañas Ayuda
noble@debian:~$ su
Contraseña:
root@debian:/home/noble# apt-get install heartbeat
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
E: No se ha podido localizar el paquete heartbeat
root@debian:/home/noble# aptitude install heartbeat
No se puede encontrar ningún paquete que coincida con "heartbeat". Sin embargo,
los siguiente paquetes contienen "heartbeat" en su descripción:
  prewikka
No se puede encontrar ningún paquete que coincida con "heartbeat". Sin embargo,
los siguiente paquetes contienen "heartbeat" en su descripción:
  prewikka
No se instalará, actualizará o eliminará ningún paquete.
0 paquetes actualizados, 0 nuevos instalados, 0 para eliminar y 69 sin actualiza
r.
Necesito descargar 0 B de ficheros. Después de desempaquetar se usarán 0 B.

root@debian:/home/noble# nano /etc/apt/sources.list
root@debian:/home/noble# apt-get install heartbeat
```

```
Archivo Editar Ver Terminal Pestañas Ayuda
GNU nano 2.2.6 Fichero: /etc/apt/sources.list

# jessie-oficiales
deb http://mirrors.kernel.org/debian/ jessie main contrib non-free
deb-src http://mirrors.kernel.org/debian/ jessie main contrib non-free

# jessie-actualizaciones-seguridad
deb http://security.debian.org/ jessie/updates main contrib non-free
deb-src http://security.debian.org/ jessie/updates main contrib non-free

# jessie-actualizaciones-sistema
deb http://mirrors.kernel.org/debian/ jessie-updates main contrib non-free
deb-src http://mirrors.kernel.org/debian/ jessie-updates main contrib non-free
deb http://mirrors.kernel.org/debian jessie-proposed-updates main contrib non-f$
deb-src http://mirrors.kernel.org/debian jessie-proposed-updates main contrib n$

# jessie-backports
deb http://mirrors.kernel.org/debian/ jessie-backports main contrib non-free
deb-src http://mirrors.kernel.org/debian/ jessie-backports main contrib non-free

[ 22 líneas leídas ]
^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U PegarTxt ^T Ortografía
```



```
Archivo Editar Ver Terminal Pestañas Ayuda
GNU nano 2.2.6          Fichero: /etc/apt/sources.list

deb http://mirrors.kernel.org/debian/ jessie-updates main contrib non-free
deb-src http://mirrors.kernel.org/debian/ jessie-updates main contrib non-free
deb http://mirrors.kernel.org/debian jessie-proposed-updates main contrib non-f$
deb-src http://mirrors.kernel.org/debian jessie-proposed-updates main contrib n$

# jessie-backports
deb http://mirrors.kernel.org/debian/ jessie-backports main contrib non-free
deb-src http://mirrors.kernel.org/debian/ jessie-backports main contrib non-free

# deb-multimedia
# apt-get install deb-multimedia-keyring
deb http://www.deb-multimedia.org jessie main non-free
█

^G Ver ayuda  ^O Guardar   ^R Leer Fich ^Y Pág Ant   ^K CortarTxt ^C Pos actual
^X Salir      ^J Justificar ^W Buscar    ^V Pág Sig   ^U PegarTxt  ^T Ortografía
```

```
Archivo Editar Ver Terminal Pestañas Ayuda
Contraseña:
root@debian:/home/noble# apt-get install heartbeat
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
E: No se ha podido localizar el paquete heartbeat
root@debian:/home/noble# aptitude install heartbeat
No se puede encontrar ningún paquete que coincida con "heartbeat". Sin embargo,
los siguiente paquetes contienen "heartbeat" en su descripción:
  prewikka
No se puede encontrar ningún paquete que coincida con "heartbeat". Sin embargo,
los siguiente paquetes contienen "heartbeat" en su descripción:
  prewikka
No se instalará, actualizará o eliminará ningún paquete.
0 paquetes actualizados, 0 nuevos instalados, 0 para eliminar y 69 sin actualiza
r.
Necesito descargar 0 B de ficheros. Después de desempaquetar se usarán 0 B.

root@debian:/home/noble# nano /etc/apt/sources.list
root@debian:/home/noble# apt-get install heartbeat
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
█
```

```
Archivo Editar Ver Terminal Pestañas Ayuda
r.
Necesito descargar 0 B de ficheros. Después de desempaquetar se usarán 0 B.

root@debian:/home/noble# nano /etc/apt/sources.list
root@debian:/home/noble# apt-get install heartbeat
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  cluster-glue gawk iproute libcurl3 libheartbeat2 liblrn2 libnet1 libopenhpi2
  libopenipmi0 libperl5.20 libpils2 libplumb2 libplumbgpl2 libsnmp-base
  libsnmp30 libstonith1 libxml2-utils openhpid resource-agents
Paquetes sugeridos:
  gawk-doc snmp-mibs-downloader
Paquetes recomendados:
  pacemaker
Se instalarán los siguientes paquetes NUEVOS:
  cluster-glue gawk heartbeat iproute libcurl3 libheartbeat2 liblrn2 libnet1
  libopenhpi2 libopenipmi0 libperl5.20 libpils2 libplumb2 libplumbgpl2
  libsnmp-base libsnmp30 libstonith1 libxml2-utils openhpid resource-agents
0 actualizados, 20 nuevos se instalarán, 0 para eliminar y 69 no actualizados.
Se necesita descargar 7,452 kB de archivos.
Se utilizarán 19.7 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

```
Archivo Editar Ver Terminal Pestañas Ayuda
¿Desea continuar? [S/n] s
Des:1 http://mirrors.kernel.org/debian/ jessie/main gawk i386 1:4.1.1+dfsg-1 [49
6 kB]
Des:2 http://mirrors.kernel.org/debian/ jessie/main libcurl3 i386 7.38.0-4+deb8u
2 [278 kB]
Des:3 http://mirrors.kernel.org/debian/ jessie/main libnet1 i386 1.1.6+dfsg-3 [6
1.8 kB]
Des:4 http://mirrors.kernel.org/debian/ jessie/main libperl5.20 i386 5.20.2-3+de
b8u1 [726 kB]
Des:5 http://mirrors.kernel.org/debian/ jessie/main libsnmp-base all 5.7.2.1+dfs
g-1 [1,544 kB]
Des:6 http://mirrors.kernel.org/debian/ jessie/main libsnmp30 i386 5.7.2.1+dfs
g-1 [2,167 kB]
Des:7 http://mirrors.kernel.org/debian/ jessie/main libpils2 i386 1.0.12-rc1+hg2
777-1.2 [24.7 kB]
Des:8 http://mirrors.kernel.org/debian/ jessie/main libplumb2 i386 1.0.12-rc1+hg
2777-1.2 [87.5 kB]
Des:9 http://mirrors.kernel.org/debian/ jessie/main liblrn2 i386 1.0.12-rc1+hg2
777-1.2 [24.6 kB]
Des:10 http://mirrors.kernel.org/debian/ jessie/main libopenhpi2 i386 2.14.1-1.4
[156 kB]
Des:11 http://mirrors.kernel.org/debian/ jessie/main libopenipmi0 i386 2.0.16-1.
4 [526 kB]
78% [11 libopenipmi0 246 kB/526 kB 47%] 294 kB/s 5s
```

```
Archivo Editar Ver Terminal Pestañas Ayuda
[156 kB]
Des:11 http://mirrors.kernel.org/debian/ jessie/main libopenipmi0 i386 2.0.16-1.4 [526 kB]
Des:12 http://mirrors.kernel.org/debian/ jessie/main libplumbgpl2 i386 1.0.12-rc1+hg2777-1.2 [14.3 kB]
Des:13 http://mirrors.kernel.org/debian/ jessie/main libstonith1 i386 1.0.12-rc1+hg2777-1.2 [21.2 kB]
Des:14 http://mirrors.kernel.org/debian/ jessie/main cluster-glue i386 1.0.12-rc1+hg2777-1.2 [276 kB]
Des:15 http://mirrors.kernel.org/debian/ jessie/main libheartbeat2 i386 1:3.0.5+hg12629-1.2 [69.7 kB]
Des:16 http://mirrors.kernel.org/debian/ jessie/main iproute all 1:3.16.0-2 [15.5 kB]
Des:17 http://mirrors.kernel.org/debian/ jessie/main libxml2-utils i386 2.9.1+dfsg1-5 [91.1 kB]
Des:18 http://mirrors.kernel.org/debian/ jessie/main resource-agents i386 1:3.9.3+git20121009-3.1 [442 kB]
Des:19 http://mirrors.kernel.org/debian/ jessie/main heartbeat i386 1:3.0.5+hg12629-1.2 [337 kB]
Des:20 http://mirrors.kernel.org/debian/ jessie/main openhpid i386 2.14.1-1.4 [94.6 kB]
Descargados 7,452 kB en 33s (223 kB/s)
Seleccionando el paquete gawk previamente no seleccionado.
Leyendo la base de datos ... 65%
```

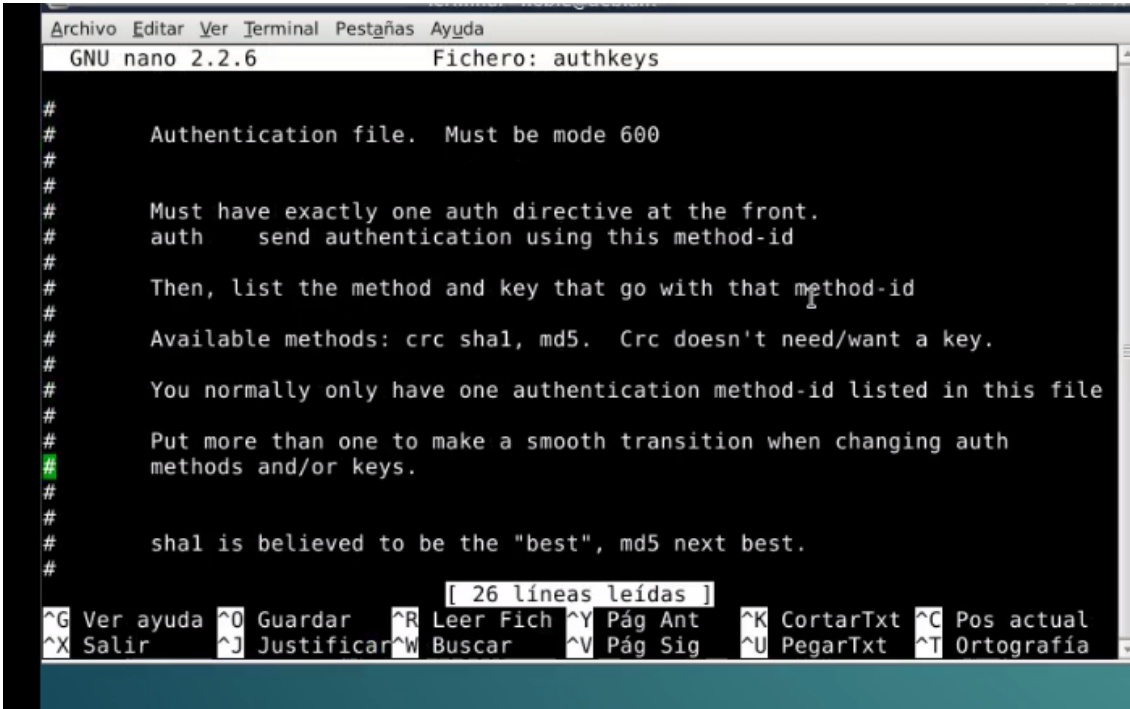
```
Archivo Editar Ver Terminal Pestañas Ayuda
Desempaquetando cluster-glue (1.0.12-rc1+hg2777-1.2) ...
Seleccionando el paquete libheartbeat2 previamente no seleccionado.
Preparando para desempaquetar .../libheartbeat2_1%3a3.0.5+hg12629-1.2_i386.deb ...
Desempaquetando libheartbeat2 (1:3.0.5+hg12629-1.2) ...
Seleccionando el paquete iproute previamente no seleccionado.
Preparando para desempaquetar .../iproute_1%3a3.16.0-2_all.deb ...
Desempaquetando iproute (1:3.16.0-2) ...
Seleccionando el paquete libxml2-utils previamente no seleccionado.
Preparando para desempaquetar .../libxml2-utils_2.9.1+dfsg1-5_i386.deb ...
Desempaquetando libxml2-utils (2.9.1+dfsg1-5) ...
Seleccionando el paquete resource-agents previamente no seleccionado.
Preparando para desempaquetar .../resource-agents_1%3a3.9.3+git20121009-3.1_i386.deb ...
Desempaquetando resource-agents (1:3.9.3+git20121009-3.1) ...
Seleccionando el paquete heartbeat previamente no seleccionado.
Preparando para desempaquetar .../heartbeat_1%3a3.0.5+hg12629-1.2_i386.deb ...
Desempaquetando heartbeat (1:3.0.5+hg12629-1.2) ...
Seleccionando el paquete openhpid previamente no seleccionado.
Preparando para desempaquetar .../openhpid_2.14.1-1.4_i386.deb ...
Desempaquetando openhpid (2.14.1-1.4) ...
Procesando disparadores para man-db (2.7.0.2-5) ...
```



```
Archivo Editar Ver Terminal Pestañas Ayuda
Procesando disparadores para libc-bin (2.19-18) ...
Procesando disparadores para systemd (215-17+deb8u1) ...
Se encontraron errores al procesar:
  openpid
E: Sub-process /usr/bin/dpkg returned an error code (1)
root@debian:/home/noble# cd /usr/share/doc
doc/      doc-base/ docutils/
root@debian:/home/noble# cd /usr/share/doc
doc/      doc-base/ docutils/
root@debian:/home/noble# cd /usr/share/doc
doc/      doc-base/ docutils/
root@debian:/home/noble# cd /usr/share/doc/h
hddtemp/      host/      hyphen-en-us/
heartbeat/    hostname/
hicolor-icon-theme/ hunspell-en-us/
root@debian:/home/noble# cd /usr/share/doc/h
hddtemp/      host/      hyphen-en-us/
heartbeat/    hostname/
hicolor-icon-theme/ hunspell-en-us/
root@debian:/home/noble# cd /usr/share/doc/heartbeat/
root@debian:/usr/share/doc/heartbeat# ls
```

```
Archivo Editar Ver Terminal Pestañas Ayuda
authkeys  changelog.Debian.gz  copyright  haresources.gz  README.cts.gz
root@debian:/usr/share/doc/heartbeat# gz -d haresources.gz
bash: gz: no se encontró la orden
root@debian:/usr/share/doc/heartbeat# gzip -d haresources.gz
root@debian:/usr/share/doc/heartbeat# ls
appbhd.cf  AUTHORS          changelog.gz  ha.cf.gz      README
authkeys  changelog.Debian.gz  copyright    haresources  README.cts.gz
root@debian:/usr/share/doc/heartbeat# cp /usr/share/doc/heartbeat/authkeys /etc/ha.d/
root@debian:/usr/share/doc/heartbeat# cp /usr/share/doc/heartbeat/haresources /etc/ha.d/
root@debian:/usr/share/doc/heartbeat# gzip /usr/share/doc/heartbeat/ha.cf.gz
gzip: /usr/share/doc/heartbeat/ha.cf.gz already has .gz suffix -- unchanged
root@debian:/usr/share/doc/heartbeat# gzip-d /usr/share/doc/heartbeat/ha.cf.gz
bash: gzip-d: no se encontró la orden
root@debian:/usr/share/doc/heartbeat# gzip -d /usr/share/doc/heartbeat/ha.cf.gz
root@debian:/usr/share/doc/heartbeat# cp /usr/share/doc/heartbeat/ha.cf /etc/ha.d/
root@debian:/usr/share/doc/heartbeat# cd /etc/ha.d/
root@debian:/etc/ha.d# ls
authkeys  ha.cf  harc  haresources  rc.d  README.config  resource.d  shellfuncs
root@debian:/etc/ha.d# nano authkeys
```

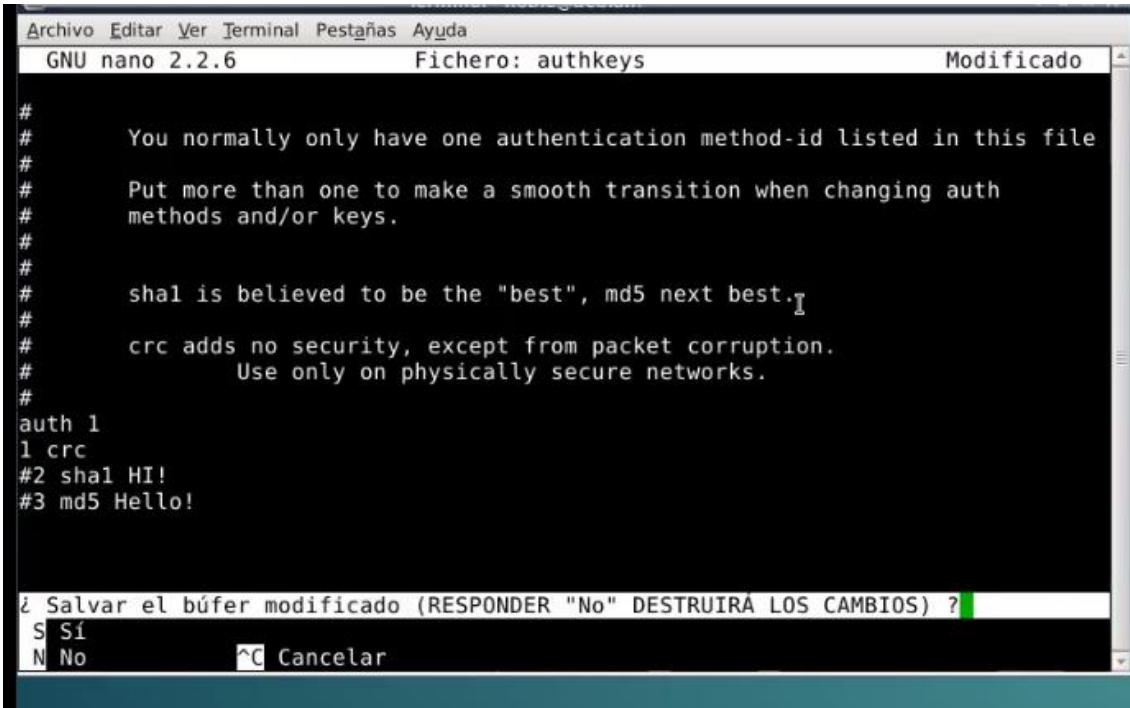
Configurando Authkeys



The screenshot shows the nano editor interface with the file 'authkeys' open. The menu bar includes 'Archivo', 'Editar', 'Ver', 'Terminal', 'Pestañas', and 'Ayuda'. The title bar shows 'GNU nano 2.2.6' and 'Fichero: authkeys'. The main text area contains the following content:

```
#
# Authentication file. Must be mode 600
#
# Must have exactly one auth directive at the front.
# auth send authentication using this method-id
#
# Then, list the method and key that go with that method-id
#
# Available methods: crc sha1, md5. Crc doesn't need/want a key.
#
# You normally only have one authentication method-id listed in this file
#
# Put more than one to make a smooth transition when changing auth
# methods and/or keys.
#
# sha1 is believed to be the "best", md5 next best.
#
```

A status bar at the bottom indicates '[26 líneas leídas]' and provides keyboard shortcuts for various actions like 'Ver ayuda', 'Guardar', 'Leer Fich', 'Pág Ant', 'CortarTxt', 'Pos actual', 'Salir', 'Justificar', 'Buscar', 'Pág Sig', 'PegarTxt', and 'Ortografía'.



The screenshot shows the nano editor interface with the file 'authkeys' open. The menu bar includes 'Archivo', 'Editar', 'Ver', 'Terminal', 'Pestañas', and 'Ayuda'. The title bar shows 'GNU nano 2.2.6', 'Fichero: authkeys', and 'Modificado'. The main text area contains the following content:

```
#
# You normally only have one authentication method-id listed in this file
#
# Put more than one to make a smooth transition when changing auth
# methods and/or keys.
#
# sha1 is believed to be the "best", md5 next best.
#
# crc adds no security, except from packet corruption.
# Use only on physically secure networks.
#
auth 1
1 crc
#2 sha1 HI!
#3 md5 Hello!
```

A confirmation dialog is displayed at the bottom: '¿ Salvar el búfer modificado (RESPONDER "No" DESTRUIRÁ LOS CAMBIOS) ?'. The options are 'S Sí', 'N No', and '^C Cancelar'.

Configurando el archivo ha.cf

```
Archivo Editar Ver Terminal Pestañas Ayuda
root@debian:/usr/share/doc/heartbeat# gz -d haresources.gz
bash: gz: no se encontró la orden
root@debian:/usr/share/doc/heartbeat# gzip -d haresources.gz
root@debian:/usr/share/doc/heartbeat# ls
apphbd.cf  AUTHORS          changelog.gz  ha.cf.gz      README
authkeys  changelog.Debian.gz  copyright     haresources  README.cts.gz
root@debian:/usr/share/doc/heartbeat# cp /usr/share/doc/heartbeat/authkeys /etc/ha.d/
root@debian:/usr/share/doc/heartbeat# cp /usr/share/doc/heartbeat/haresources /etc/ha.d/
root@debian:/usr/share/doc/heartbeat# gzip /usr/share/doc/heartbeat/ha.cf.gz
gzip: /usr/share/doc/heartbeat/ha.cf.gz already has .gz suffix -- unchanged
root@debian:/usr/share/doc/heartbeat# gzip-d /usr/share/doc/heartbeat/ha.cf.gz

bash: gzip-d: no se encontró la orden
root@debian:/usr/share/doc/heartbeat# gzip -d /usr/share/doc/heartbeat/ha.cf.gz

root@debian:/usr/share/doc/heartbeat# cp /usr/share/doc/heartbeat/ha.cf /etc/ha.d/
root@debian:/usr/share/doc/heartbeat# cd /etc/ha.d/
root@debian:/etc/ha.d# ls
authkeys ha.cf  harc  haresources  rc.d  README.config  resource.d  shellfuncs
root@debian:/etc/ha.d# nano authkeys
root@debian:/etc/ha.d# nano ha.cf
```

```
Archivo Editar Ver Terminal Pestañas Ayuda
GNU nano 2.2.6 Fichero: ha.cf
#
# There are lots of options in this file. All you have to have is a set
# of nodes listed {"node ...} one of {serial, bcst, mcast, or ucast},
# and a value for "auto_failback".
#
# ATTENTION: As the configuration file is read line by line,
# THE ORDER OF DIRECTIVE MATTERS!
#
# In particular, make sure that the udpport, serial baud rate
# etc. are set before the heartbeat media are defined!
# debug and log file directives go into effect when they
# are encountered.
#
# All will be fine if you keep them ordered as in this example.
#
# Note on logging:
# If all of debugfile, logfile and logfacility are not defined,
# logging is the same as use logd yes. In other case, they are
#
[ 340 líneas leídas ]
^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U PegarTxt ^T Ortografia
```



```
Archivo Editar Ver Terminal Pestañas Ayuda
GNU nano 2.2.6 Fichero: ha.cf Modificado
#
# If you set this too low you will get the problematic
# split-brain (or cluster partition) problem.
# See the FAQ for how to use warntime to tune deadtime.
#
deadtime 30
#
# warntime: how long before issuing "late heartbeat" warning?
# See the FAQ for how to use warntime to tune deadtime.
#
warntime 10
#
#
# Very first dead time (initdead)
#
# On some machines/OSes, etc. the network takes a while to come up
# and start working right after you've been rebooted. As a result
# we have a separate dead time for when things first come up.
# It should be at least twice the normal dead time.
#
^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U PegarTxt ^T Ortografía
```

```
Archivo Editar Ver Terminal Pestañas Ayuda
GNU nano 2.2.6 Fichero: ha.cf Modificado
ucast eth0 192.168.0.21
#
#
# About boolean values...
#
# Any of the following case-insensitive values will work for true:
# true, on, yes, y, 1
# Any of the following case-insensitive values will work for false:
# false, off, no, n, 0
#
#
# auto_failback: determines whether a resource will
# automatically fail back to its "primary" node, or remain
# on whatever node is serving it until that node fails, or
# an administrator intervenes.
#
# The possible values for auto failback are:
# on - enable automatic failbacks
#
^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U PegarTxt ^T Ortografía
```



```
Archivo Editar Ver Terminal Pestañas Ayuda
GNU nano 2.2.6 Fichero: ha.cf Modificado

# NOTE: If you are using the software watchdog, you very likely
# wish to load the module with the parameter "nowayout=0" or
# compile it without CONFIG_WATCHDOG_NOWAYOUT set. Otherwise even
# an orderly shutdown of heartbeat will trigger a reboot, which is
# very likely NOT what you want.
#
#watchdog /dev/watchdog
#
# Tell what machines are in the cluster
# node nodename ... -- must match uname -n
node pi
node debian
#
# Less common options...
#
# Treats 10.10.10.254 as a psuedo-cluster-member
# Used together with ipfail below...
# note: don't use a cluster node as ping node
#
^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U PegarTxt ^T Ortografía
```

Configuración del archivo haresources

```
Archivo Editar Ver Terminal Pestañas Ayuda
bash: gz: no se encontró la orden
root@debian:/usr/share/doc/heartbeat# gzip -d haresources.gz
root@debian:/usr/share/doc/heartbeat# ls
apphbd.cf AUTHORS changelog.gz ha.cf.gz README
authkeys changelog.Debian.gz copyright haresources README.cts.gz
root@debian:/usr/share/doc/heartbeat# cp /usr/share/doc/heartbeat/authkeys /etc/
ha.d/
root@debian:/usr/share/doc/heartbeat# cp /usr/share/doc/heartbeat/haresources /e
tc/ha.d/
root@debian:/usr/share/doc/heartbeat# gzip /usr/share/doc/heartbeat/ha.cf.gz
gzip: /usr/share/doc/heartbeat/ha.cf.gz already has .gz suffix -- unchanged
root@debian:/usr/share/doc/heartbeat# gzip-d /usr/share/doc/heartbeat/ha.cf.gz

bash: gzip-d: no se encontró la orden
root@debian:/usr/share/doc/heartbeat# gzip -d /usr/share/doc/heartbeat/ha.cf.gz

root@debian:/usr/share/doc/heartbeat# cp /usr/share/doc/heartbeat/ha.cf /etc/ha.
d/
root@debian:/usr/share/doc/heartbeat# cd /etc/ha.d/
root@debian:/etc/ha.d# ls
authkeys ha.cf harc haresources rc.d README.config resource.d shellfuncs
root@debian:/etc/ha.d# nano authkeys
root@debian:/etc/ha.d# nano ha.cf
root@debian:/etc/ha.d# nano haresources
```

```
Archivo Editar Ver Terminal Pestañas Ayuda
GNU nano 2.2.6 Fichero: haresources Modificado

pi IPadrrs::192.168.0.20/24/eth0/vsftpd
#
# This is a list of resources that move from machine to machine as
# nodes go down and come up in the cluster. Do not include
# "administrative" or fixed IP addresses in this file.
#
# <VERY IMPORTANT NOTE>
# The haresources files MUST BE IDENTICAL on all nodes of the cluster.
#
# The node names listed in front of the resource group information
# is the name of the preferred node to run the service. It is
# not necessarily the name of the current machine. If you are running
# auto_failback ON (or legacy), then these services will be started
# up on the preferred nodes - any time they're up.
#
# If you are running with auto_failback OFF, then the node information
# will be used in the case of a simultaneous start-up, or when using
# the hb standby {foreign,local} command.
Nombre del fichero a escribir: haresources
^G Ver ayuda M-D Formato DOS M-A Añadir M-B Respalda fich
^C Cancelar M-M Formato Mac M-P Anteponer
```

6.2 Instalación de vsftpd

```
Archivo Editar Ver Terminal Pestañas Ayuda
service failed. See 'systemctl status heartbeat.service' and 'journalctl -xn' fo
r details.
failed!
root@debian:/etc/ha.d# nano ha.cf
root@debian:/etc/ha.d# apt-get install vsftpd
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  dialog
Se instalarán los siguientes paquetes NUEVOS:
  dialog vsftpd
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 69 no actualizados.
1 no instalados del todo o eliminados.
Se necesita descargar 424 kB de archivos.
Se utilizarán 1,201 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://mirrors.kernel.org/debian/ jessie/main dialog i386 1.2-20140911-1 [
264 kB]
Des:2 http://mirrors.kernel.org/debian/ jessie/main vsftpd i386 3.0.2-17 [160 kB
]
Descargados 424 kB en 2s (176 kB/s)
Preconfigurando paquetes ...
```

```
Archivo Editar Ver Terminal Pestañas Ayuda
Desempaquetando vsftpd (3.0.2-17) ...
Procesando disparadores para man-db (2.7.0.2-5) ...
Procesando disparadores para systemd (215-17+deb8u1) ...
Configurando openhpid (2.14.1-1.4) ...
Job for openhpid.service failed. See 'systemctl status openhpid.service' and 'journalctl -xn' for details.
invoke-rc.d: initscript openhpid, action "start" failed.
dpkg: error al procesar el paquete openhpid (--configure):
 el subproceso instalado el script post-installation devolvió el código de salida de error 1
Configurando dialog (1.2-20140911-1) ...
Configurando vsftpd (3.0.2-17) ...
Procesando disparadores para systemd (215-17+deb8u1) ...
Se encontraron errores al procesar:
openhpid
E: Sub-process /usr/bin/dpkg returned an error code (1)
root@debian:/etc/ha.d# /etc/init.d/vsftpd status
● vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled)
   Active: active (running) since mié 2015-11-18 10:08:34 CST; 26s ago
 Main PID: 5974 (vsftpd)
   CGroup: /system.slice/vsftpd.service
           └─5974 /usr/sbin/vsftpd /etc/vsftpd.conf
root@debian:/etc/ha.d# /etc/init.d/vsftpd status
```

```
terminal - noble@debian:
GNU nano 2.2.6 Fichero: /etc/vsftpd.conf Modificado
#
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=YES
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (:::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U PegarTxt ^T Ortografía
```



```
Archivo Editar Ver Terminal Pestañas Ayuda
GNU nano 2.2.6 Fichero: /etc/vsftpd.conf Modificado

write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
anon_mkdir_write_enable=YES
#
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
#
^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U PegarTxt ^T Ortografía
```

6.3 Instalación de Incron

```
Archivo Editar Ver Terminal Pestañas Ayuda
CGroup: /system.slice/vsftpd.service
└─5974 /usr/sbin/vsftpd /etc/vsftpd.conf
root@debian:/etc/ha.d# nano /etc/vsftpd.conf
root@debian:/etc/ha.d# apt-get install incron
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
incron
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 69 no actualizados.
1 no instalados del todo o eliminados.
Se necesita descargar 68.2 kB de archivos.
Se utilizarán 222 kB de espacio de disco adicional después de esta operación.
Des:1 http://mirrors.kernel.org/debian/ jessie/main incron i386 0.5.10-2 [68.2 kB]
Descargados 68.2 kB en 0s (113 kB/s)
Seleccionando el paquete incron previamente no seleccionado.
(Leyendo la base de datos ... 109509 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../incron_0.5.10-2_i386.deb ...
Desempaquetando incron (0.5.10-2) ...
Procesando disparadores para man-db (2.7.0.2-5) ...
Procesando disparadores para systemd (215-17+deb8u1) ...
```

```

Archivo Editar Ver Terminal Pestañas Ayuda
incron.d/      init.d/        insserv/
init/          initramfs-tools/ insserv.conf.d/
root@debian:/etc/ha.d# /etc/in
incron.d/      init.d/        insserv/
init/          initramfs-tools/ insserv.conf.d/
root@debian:/etc/ha.d# /etc/in
incron.d/      init.d/        insserv/
init/          initramfs-tools/ insserv.conf.d/
root@debian:/etc/ha.d# /etc/in
incron.d/      init.d/        insserv/
init/          initramfs-tools/ insserv.conf.d/
root@debian:/etc/ha.d# nano /etc/in
incron.allow  incron.deny    initramfs-tools/ insserv.conf
incron.conf   init/          inputrc          insserv.conf.d/
incron.d/     init.d/        insserv/
root@debian:/etc/ha.d# nano /etc/in
incron.allow  incron.deny    initramfs-tools/ insserv.conf
incron.conf   init/          inputrc          insserv.conf.d/
incron.d/     init.d/        insserv/
root@debian:/etc/ha.d# nano /etc/incron.allow
root@debian:/etc/ha.d# █

```

6.4 Sincronización de archivos

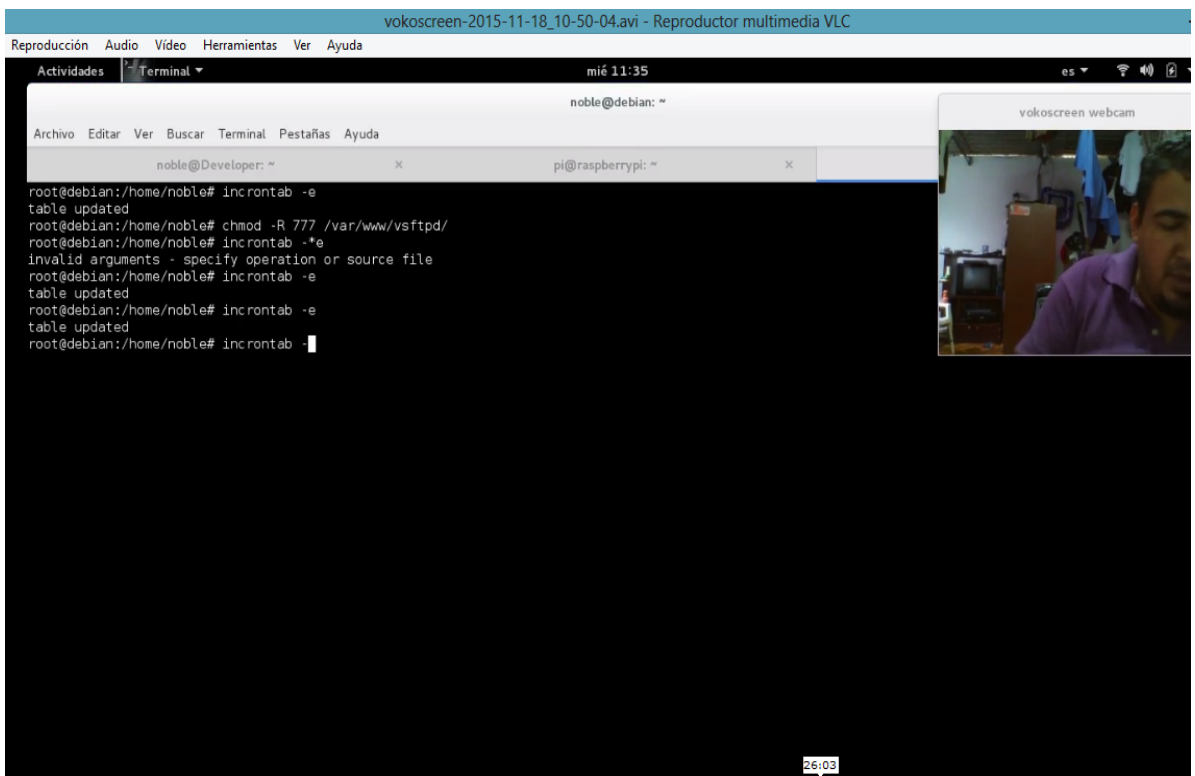
```

Terminal - noble@debian: ~
GNU nano 2.2.6      Fichero: /tmp/incron.table-8F03WB      Modificado
/var/www/vsftpd/ IN_CREATE, IN_MODIFY, IN_MOVED_TO rsync █var/www/vsftpd/ debia$

I

^G Ver ayuda  ^O Guardar   ^R Leer Fich ^Y Pág Ant   ^K CortarTxt ^C Pos actual
^X Salir      ^J Justificar ^W Buscar    ^V Pág Sig   ^U PegarTxt  ^T Ortografia

```



vokoscreen-2015-11-18_10-50-04.avi - Reproductor multimedia VLC

Medio Reproducción Audio Vídeo Herramientas Ver Ayuda

Actividades Navegador web Firefox mié 11:35

Tutorial de "incron" | Cyberhades - Mozilla Firefox

Index of ftp://pedro... x Problem loading page x Tutorial de "incron" | ... x es/NetworkConfigura... x

www.cyberhades.com/2009/01/30/tutorial-de-incron/

Más visitados Getting Started

IN_ACCESS: Un fichero es accedido
 IN_MODIFY: Un fichero es modificado.
 IN_ATTRIB: Cambio en los Metadatos (permisos, atributos, fecha, etc.)
 IN_CLOSE_WRITE: Fichero que se abrió en modo escritura ha sido cerrado.
 IN_CLOSE_NOWRITE: Fichero que se no abrió en modo escritura ha sido cerrado.
 IN_OPEN: Se ha abierto un fichero.
 IN_MOVED_FROM: Fichero ha sido movido fuera del directorio que estamos monitorizando.
 IN_MOVED_TO: Fichero se ha movido al directorio que estamos monitorizando.
 IN_CREATE: Creación de un nuevo fichero o directorio en el directorio que estamos monitorizando.
 IN_DELETE: Un fichero o directorio ha sido borrado en el directorio monitorizado.
 IN_DELETE_SELF: El fichero o directorio que estaba siendo monitorizado ha sido borrado.
 IN_CLOSE: Incluye los eventos IN_CLOSE_WRITE y IN_CLOSE_NOWRITE.
 IN_MOVE: Incluye los eventos IN_MOVED_FROM y IN_MOVED_TO
 IN_ALL_EVENTS: Monitoriza todos los eventos.
 IN_DONT_FOLLOW: No desreferencia el path si es un enlace simbólico.
 IN_ONLYDIR: Sólo monitoriza el path si es un directorio
 IN_MOVE_SELF: Cuando el fichero o directorio es movido.
 IN_ONESHOT: Monitoriza el path para un sólo evento.

incrontab -e : nos permite editar las reglas.
 incrontab -f : borra la tabla de reglas por completo. OJO, ¡NO PIDE CONFIRMACION!
 incrontab -d : recarga la tabla de reglas. Cuando modifiquemos nuestra tabla de reglas, para que se haga efectivo, tenemos que ejecutar este comando.

incrontab -u : ejecuta incrontab como otro usuario. La intención de esta opción es el poder modificar la tabla de reglas de los usuarios del sistema que no suelen tener shell, como apache, postfix, etc. Solo aquellos usuarios con privilegios de root pueden hacer uso de esta opción.

incrontab -f : nos permite indicarle un fichero de configuración. Por defecto usa /etc/incron.conf. En este fichero entre otras cosas personalizar el editor de texto usado cuando ejecutamos incrontab -e. También

0
 0
 3
 0
 0
 0

Publicaciones más comentadas

Como construir un inhibidor de teléfonos móviles casero - 31 comentarios

26:09

vokoscreen-2015-11-18_10-50-04.avi - Reproductor multimedia VLC

producción Audio Vídeo Herramientas Ver Ayuda

Actividades Navegador web Firefox mié 11:37

Index of ftp://pedro@192.168.0.21/ - Mozilla Firefox

Index of ftp://pedro... x Tutorial de "incron" | ... x es/NetworkConfigura... x Index of ftp://pedro... x

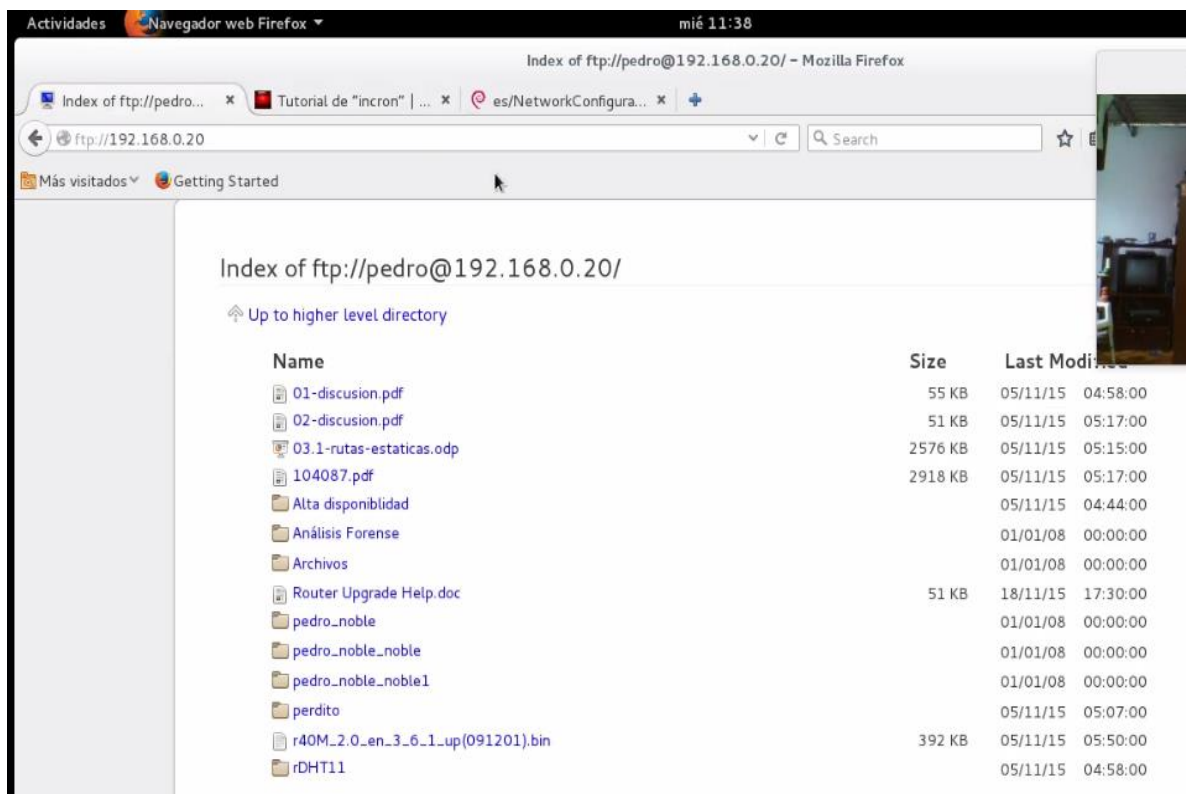
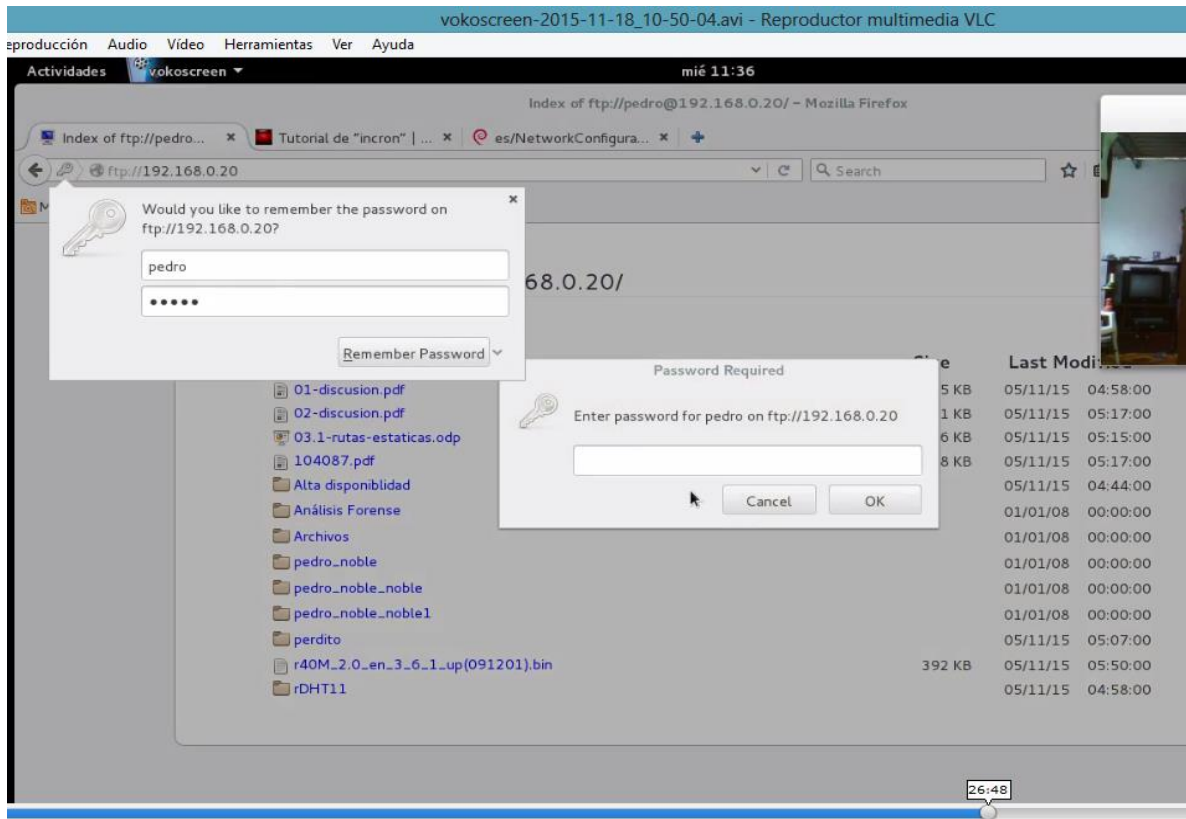
ftp://192.168.0.21/

Más visitados Getting Started

Index of ftp://pedro@192.168.0.21/

Up to higher level directory

Name	Size	Last Modified
01-discusion.pdf	55 KB	04/11/15 22:58:00
02-discusion.pdf	51 KB	04/11/15 23:17:00
03.1-rutas-estaticas.odp	2576 KB	04/11/15 23:15:00
104087.pdf	2918 KB	04/11/15 23:17:00
Alta disponibilidad		04/11/15 22:44:00
Análisis Forense		31/12/07 00:00:00
Archivos		31/12/07 00:00:00
Router Upgrade Help.doc	51 KB	18/11/15 11:30:00
pedro_noble		31/12/07 00:00:00
pedro_noble_noble		31/12/07 00:00:00
pedro_noble_noble1		31/12/07 00:00:00
perdido		04/11/15 23:07:00
rDHT11		04/11/15 22:58:00



7 CONCLUSIONES

Con la realización de este proyecto hemos logrado comprender el procedimiento que se lleva a cabo para la configuración de Heartbeat, así como su funcionamiento, el cual garantiza la alta disponibilidad en los servicios que se ofrecen en los diferentes entornos laborales o empresariales, permitiendo una mejor atención a los usuarios quienes no se dan cuenta del fallo de alguno de los servidores, dado a la rapidez con que se logra restablecer el servicio al pasar a funcionar el servidor de respaldo, todo esto gracias al monitoreo automatizado que realiza Heartbeat, a través de una especie de latidos o ping que vienen siendo mensajes ICMP, los cuales verifican o controlan el funcionamiento de los servidores de tal manera que cuando uno de ellos falla, pasa a funcionar el otro. De esta manera se garantiza una alta disponibilidad en los servicios que deben estar activos de manera permanente para ser brindados a los clientes y que hoy en día son demandados con mayor exigencia.

8 BIBLIOGRAFIA

- ✓ <http://es.slideshare.net/josealbalat/heartbeat-11717697>
- ✓ <http://blog-alexis.rhcloud.com/2011/08/18/alta-disponibilidad-con-heartbeat/>
- ✓ <http://www.alcancelibre.org/staticpages/index.php/como-cluster-heartbeat-centos>
- ✓ <http://blog-alexis.rhcloud.com/2011/08/18/alta-disponibilidad-con-heartbeat/>

ANEXOS

IMÁGENES DEL EQUIPO UTILIZADO PARA LA REALIZACION DEL PROYECTO.



Raspberry pi utilizada como servidor



Laptop utilizada como cliente



Mini laptop haciendo la función del segundo servidor para el cluster



Router, dispositivo que nos proporciona la conectividad entre los nodos