

**Universidad Luterana Salvadoreña.
Facultad de Ciencias del Hombre y la Naturaleza.
Licenciatura en Ciencias de la Computación.**



ESPECIALIZACIÓN EN ADMINISTRACIÓN DE SERVIDORES LINUX

**TEMA:
CLUSTER DE ALTA DISPONIBILIDAD CON APACHE2 Y HEARTBEAT
EVALUACIÓN PROYECTO FINAL.**

Ingeniero: Manuel de Jesús Flores Villatoro.

Integrantes del grupo.

N°	Apellido	Nombre	Carnet
1	Flores Martinez	José Manuel	FM02110857
2	García Pineda	Osiris Stephanie	GP02110201
3	Mazana Escobar	Manuel Antonio	ME01121555

Fecha de entrega: San Salvador, 20 de enero de 2018.

INDICE

Introducción.....	1
Objetivos del proyecto.....	2
Marco Teórico.....	3
¿Qué es un clúster?.....	3
Elementos que forman parte de un clúster.....	3
Tipos de configuración.....	6
Activo/Pasivo.....	6
Conexión de red.....	7
Web server:.....	8
¿Cómo funciona un clúster?.....	10
Software para la implementación de un clúster.....	11
¿Por qué es necesaria la implementación de un clúster?.....	12
Apache.....	14
¿Qué es DRBD?.....	14
Descripción del proyecto:.....	15
Requisitos necesarios para la implementación del proyecto.....	15
Sección de términos.....	16
Descripción del proyecto final.....	17
Diagrama de GANTT.....	18
Diagrama de red.....	19
Presupuesto.....	20
Viabilidad.....	21
Ventajas de un clúster web de alta disponibilidad.....	22
Desventajas.....	22
Desarrollo del prototipo.....	23
Instalación de las maquinas virtuales.....	23
Buenas prácticas para la construcción del proyecto.....	24
Conclusión.....	25
La configuración de un clúster que funcione como un servidor de alta disponibilidad, nos ha permitido aprender sobre que software utilizar para lograrlo, y vencer los obstáculos que en el campo de las pruebas fueron apareciendo, así como solventar los errores que cometimos al crear los nodos virtuales y las configuraciones necesarias para lograr el objetivo.....	25
Recomendaciones.....	26
Anexos.....	27
Instalación y preparación de MV (maquinas virtuales).....	27
Instalación de Apache en MV (maquinas virtuales).....	30
Instalación de heartbeat.....	31
Configuración de heartbeat.....	33
Bibliografía.....	50

Introducción.

El Presente proyecto tiene como fin mostrar el funcionamiento de un **“clúster web con alta disponibilidad con replicación de archivos Apache2, heartbeat y DRBD”**. Dicho proyecto se configuro e implemento como prueba haciendo uso de tres maquinas virtuales con Oracle VM VirtualBox y a la vez se puede presentar dicha implementación en maquinas físicas. Para ello se configuraron dos nodos “servidores” y un nodo “cliente” que constituye el clúster el cual tendrá la disponibilidad de mantener ambos nodos activo activo.

Se presentan las diferentes tecnologías asociadas, una descripción de lo que se pretende implementar, así como un estudio básico de factibilidades.

Un clúster es un conjunto de computadoras que trabajan de manera colectiva, compartiendo recursos entre sí, para operar como una sola máquina. Su trabajo permitirá que siempre este disponible, ya que si un servicio falla entrara en actividad otro de tal forma que siempre el servicio este trabajando según la configuración que es activo activo.

La alta disponibilidad hace referencia a que un sistema debe estar funcionando las 24 horas del día los 365 días del año. Para poder lograr estas características es necesario el uso de técnicas de balanceo de carga y de recuperación ante fallos.

¿Cual es el propósito de tener un clúster web?

Como sabemos hoy en día los sitios web son parte fundamental de las empresas, en el área de marketing para dirigir publicidad, promocionar servicios, entre otras cosas. también se utilizan para almacenar información de consulta constante para los empleados. Muchas personas en todo el mundo pueden estar realizando solicitudes a un servidor para poder visualizar una misma página web, con tantas peticiones es común que el tiempo de respuesta en algún momento del día sea prolongado, incluso puede que la página esté inaccesible por una cierta cantidad de tiempo. Es por eso que trataremos de resolver este inconveniente en nuestro servidor para que así esté disponible en línea la mayor parte del tiempo posible y que el usuario final pase desapercibido casi cualquier falla presentada en el servidor, además utilizando las bondades de los sistemas de multiprocesamiento evitaremos cuellos de botella en las peticiones al servidor, usando para ello un balanceo de carga, logrando así gestionar las solicitudes de usuarios al servidor, con posibilidad a ser un sistema escalable en cuanto se requiera.

Objetivos del proyecto.

Objetivos Generales

- Desarrollar un cluster web de alta disponibilidad , con Apache2; heartbeat y DRBD. .

Objetivos Específicos

- Mostrar la funcionalidad de heartbeat.
- Presentar la replicación de archivos en DRBD.
- Mostrar paso a paso los comandos y las herramientas a utilizar para el montaje del proyecto.
- Realizar pruebas de comprobación que dicho proyecto esta funcionando.

Marco Teórico

CLÚSTER DE SERVIDOR DE FICHEROS DE ALTA DISPONIBILIDAD.

¿Qué es un clúster?

Es un conjunto de computadoras construidas mediante la utilización de componentes de hardware que se comportan como si fuesen una única computadora..

Elementos que forman parte de un clúster

Los elementos con los que cuenta un clúster son:

- Un nodo activo, donde corren los servicios
- Un nodo pasivo que funciona como respaldo (Backup).
- Servidores reales.
- Software de administración.
- Protocolos de comunicación y servicios.
- Conexiones de red.
- Ambientes de programación paralela.

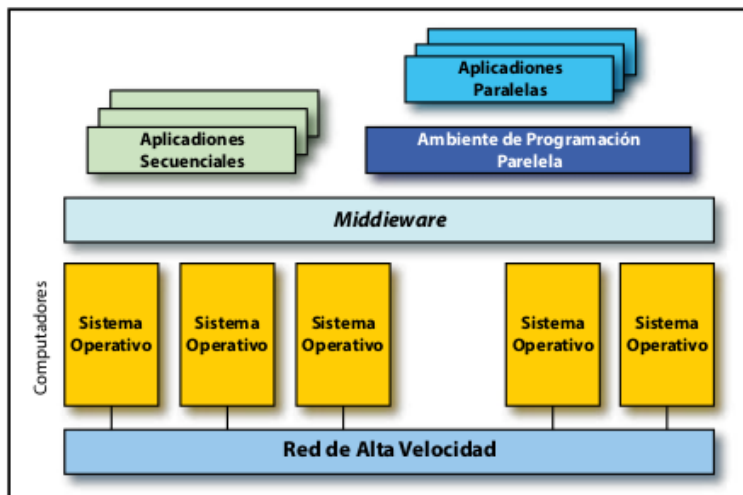


Figura 1. Representación gráfica de los elementos de un clúster.

Componentes de un clúster

En general, un clúster necesita de varios componentes de software y hardware para poder funcionar:

Nodos:

Pueden ser simples ordenadores, sistemas multiprocesador o estaciones de trabajo (workstations). En informática, de forma muy general, un nodo es un punto de intersección o unión de varios elementos que confluyen en el mismo lugar. Ahora bien, dentro de la informática la palabra nodo puede referirse a conceptos diferentes según el ámbito en el que nos movamos:

- En redes de computadoras cada una de las máquinas es un nodo, y si la red es Internet, cada servidor constituye también un nodo.
- En estructuras de datos dinámicas un nodo es un registro que contiene un dato de interés y al menos un puntero para referenciar (apuntar) a otro nodo. Si la estructura tiene solo un puntero, la única estructura que se puede construir con él es una lista, si el nodo tiene más de un puntero ya se pueden construir estructuras más complejas como árboles o grafos.

Almacenamiento:

El almacenamiento puede consistir en una NAS, una SAN, o almacenamiento interno en el servidor. El protocolo más comúnmente utilizado es NFS (Network File System), sistema de ficheros compartido entre servidor y los nodos. Sin embargo existen sistemas de ficheros específicos para clústeres como clúster(CFS) y PVFS2.

Tecnologías en el soporte del almacenamiento en discos duros:

- IDE o ATA: velocidades de 33, 66, 100, 133 y 166 MB/s
- SATA: velocidades de 150, 300 y 600 MB/s
- SCSI: velocidades de 160, 320, 640 MB/s. Proporciona altos rendimientos.
- SAS: aúna SATA-II y SCSI. Velocidades de 300 y 600 MB/s

Las unidades de cinta (DLT) son utilizadas para copias de seguridad por su bajo costo.

NAS (Network Attached Storage) es un dispositivo específico dedicado al almacenamiento a través de red (normalmente TCP/IP) que hace uso de un sistema operativo optimizado para dar acceso a través de protocolos CIFS, NFS, FTP o TFTP.

Por su parte, DAS (Direct Attached Storage) consiste en conectar unidades externas de almacenamiento SCSI o a una SAN (storage area network: 'red de área de almacenamiento') a través

de un canal de fibra. Estas conexiones son dedicadas. Mientras NAS permite compartir el almacenamiento, utilizar la red, y tiene una gestión más sencilla, DAS proporciona mayor rendimiento y mayor fiabilidad al no compartir el recurso.

Sistema operativo.

Un sistema operativo debe ser multiproceso y multiusuario. Otras características deseables son la facilidad de uso y acceso. Un sistema operativo es un programa o conjunto de programas de computadora destinado a permitir una gestión eficaz y segura de sus recursos. Comienza a trabajar cuando el gestor de arranque carga en memoria su núcleo y gestiona el hardware de la máquina desde los niveles más básicos, permitiendo también la interacción con el usuario. Se puede encontrar normalmente en la mayoría de los aparatos electrónicos que utilicen microprocesadores para funcionar, ya que gracias a estos podemos entender la máquina y que ésta cumpla con sus funciones (teléfonos móviles, reproductores de DVD, radios, computadoras, etc.).

Ejemplos:

- GNU/Linux
- ABC GNU/Linux1
- OpenMosix
- Rocks2
- Kerrighed

- Cónдор
- Sun Grid Engine
- Unix
- Solaris
- HP-UX
- AIX
- Windows
- NT

- 2000 Server
- 2003 Server
- 2008 Server
- Mac OS X
- Xgrid
- Solaris
- FreeBSD

Ambientes de Programación Paralela:

Los ambientes de programación paralela permiten implementar algoritmos que hagan uso de **recursos compartidos**: CPU (central processing unit: ‘unidad central de proceso’), memoria, datos y servicios.

Tipos de configuración

El tamaño más habitual de un clúster HA es de dos nodos, ya que es el mínimo exigido para disponer de redundancia. Las dos configuraciones más comunes en los clústers de dos nodos son:

Cluster Activo/Activo

- Aprovecha mucho mejor los recursos físicos.
- Permite repartir mejor la carga entre los nodos.

Activo/Pasivo

- Configuración más sencilla

Un clúster de alta disponibilidad es un conjunto de dos o más máquinas que se caracterizan por mantener una serie de servicios compartidos y por estar constantemente monitorizándose entre sí. Podemos dividirlo en dos clases:

Alta disponibilidad de infraestructura: Si se produce un fallo de hardware en alguna de las máquinas del clúster, el software de alta disponibilidad es capaz de arrancar automáticamente los servicios en cualquiera de las otras máquinas del clúster (failover). Y cuando la máquina que ha fallado se recupera, los servicios son nuevamente migrados a la máquina original (failback). Esta capacidad de recuperación automática de servicios nos garantiza la alta disponibilidad de los servicios ofrecidos por el clúster, minimizando así la percepción del fallo por parte de los usuarios.

Alta disponibilidad de aplicación: Si se produce un fallo del hardware o de las aplicaciones de alguna de las máquinas del clúster, el software de alta disponibilidad es capaz de arrancar automáticamente los servicios que han fallado en cualquiera de las otras máquinas del clúster. Y cuando la máquina que ha fallado se recupera, los servicios son nuevamente migrados a la máquina original. Esta capacidad de recuperación automática de servicios nos garantiza la integridad de la información, ya que no hay pérdida de datos, y además evita molestias a los usuarios, que no tienen por qué notar que se ha producido un problema.

No hay que confundir un clúster de alta disponibilidad con un clúster de alto rendimiento. El segundo es una configuración de equipos diseñado para proporcionar capacidades de cálculo mucho mayores que la que proporcionan los equipos individuales.

el primer tipo de clúster está diseñado para garantizar el funcionamiento ininterrumpido de ciertas aplicaciones.

Servicios Utilizados:

1- Heartbeat: es un Servicio(Demonio) que proporciona funcionalidad de infraestructura de clúster (comunicación y pertenencia) a sus clientes.

Esto permite a los clientes tener conocimiento de la presencia (o desaparición) de los procesos en otras maquinas e intercambiar fácilmente mensajes entre ellos.

Para resultar útil a los usuarios el dominio HEARTBEAT necesita emplearse en combinación con un gestor de recursos del cluster (cluster resource manager (CRM)) el cual posee la tarea de iniciar y parar los servicios (Direcciones IP, servidores web...) a los cuales el cluster aportará alta disponibilidad. Pacemaker es el gestor de recursos de clúster preferido para los clústers basados en Heartbeat.

Para que sirva: Heartbeat es el Encargado de revisar que cada nodo se halle funcionando. En caso un nodo falle migrará los recursos a otro nodo que también se halle ejecutando el servicio heartbeat.

2- Pacemaker: Verifica el estado de los recursos (o servicios) que le han sido asignados. En caso algún servicio falle, en caso se halla configurado, Pacemaker puede reiniciarlo. Mientras heartbeat se encarga de revisar el estado de cada nodo; Pacemaker es el responsable de verificar el estado de los servicios que deseemos sean HA dentro de los nodos.

Conexión de red

Los nodos de un clúster pueden conectarse mediante una simple red Ethernet con placas comunes (adaptadores de red o NICs), o utilizarse tecnologías especiales de alta velocidad como Fast Ethernet, Gigabit Ethernet, Myrinet, InfiniBand, SCI, etc.

- Ethernet

Son las redes más utilizadas en la actualidad, debido a su relativo bajo coste. No obstante, su tecnología limita el tamaño de paquete, realizan excesivas comprobaciones de error y sus protocolos no son eficientes, y sus velocidades de transmisión pueden limitar el rendimiento de los clústeres. Para aplicaciones con paralelismo de grano grueso puede suponer una solución acertada.

Middleware:

El middleware es un software que generalmente actúa entre el sistema operativo y las aplicaciones con la finalidad de proveer a un clúster lo siguiente:

- Una interfaz única de acceso al sistema, denominada SSI (Single System Image), la cual genera la sensación al usuario de que utiliza un único ordenador muy potente; herramientas para la optimización y mantenimiento del sistema: migración de procesos, checkpoint-restart (congelar uno o varios procesos, mudarlos de servidor y continuar su funcionamiento en el nuevo host), balanceo de carga, tolerancia a fallos, etc.
- Escalabilidad: debe poder detectar automáticamente nuevos servidores conectados al clúster para proceder a su utilización.

Web server:

Básicamente, un servidor web sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que hablan el uno con el otro mediante el protocolo de HTTP. Se pueden utilizar varias tecnologías en el servidor para aumentar su potencia más allá de su capacidad de entregar páginas HTML; éstas incluyen scripts CGI, seguridad SSL y páginas activas del servidor (ASP).

Clasificación de los clústeres.

Los clústeres pueden clasificarse con base en sus características. Hay clústeres de alto rendimiento o HighPerformance Clúster (HPC), clústeres de alta disponibilidad o High Availability (HA) y clústeres de alta eficiencia o High Throughput (HT) .

High performance: Son clústeres en los cuales se ejecutan tareas que requieren una gran capacidad computacional, cantidades enormes de memoria o ambas a la vez. Llevar a cabo estas tareas puede comprometer los recursos del clúster por largos periodos.

High availability: Son clústeres cuyo objetivo es proveer disponibilidad y confiabilidad. Estos clústeres tratan de brindar la máxima disponibilidad de los servicios que ofrecen. La confiabilidad se provee mediante un software que detecta fallos y permite recuperarse frente a ellos, mientras que en hardware se evita tener un único punto de fallos.

High throughput: Son clústeres cuyo objetivo de diseño es ejecutar la mayor cantidad de tareas en el menor tiempo posible; existe independencia de datos entre las tareas individuales. El retardo entre los nodos del clúster no es considerado un gran problema.

Balaneo de carga: Clúster que permite que un conjunto de servidores compartan la carga de trabajo y de tráfico a sus clientes. Está compuesto por uno o más ordenadores (llamados nodos) que actúan como front-end del clúster y se ocupa de repartir las peticiones de servicio que reciba el clúster a otros ordenadores que forman su back-end.

Las características más destacadas de este tipo de clúster son:

- Se puede ampliar su capacidad fácilmente añadiendo más ordenadores al clúster.
- Robustez. Ante la caída de alguno de los ordenadores del clúster, el servicio se puede ver mermado; pero mientras haya ordenadores en funcionamiento estos seguirán dando el servicio.

¿Cómo funciona un clúster?

Desde un punto de vista general, un clúster consta de dos partes. La primera es el software, un sistema operativo confeccionado especialmente para esta tarea (por ejemplo un Kernel Linux modificado). Luego se tienen compiladores y aplicaciones especiales que permiten que los programas que se ejecuten en el sistema utilicen todas las ventajas del clúster. En el entorno de GNU/Linux hay que destacar la PVM (Paralell Virtual Machine) y la MPI (Message Passing Interface), librerías que abstraen el componente hardware del componente software.

El segundo componente es la interconexión hardware entre las máquinas (nodos) del clúster. Se han desarrollado interfaces de interconexión especiales muy eficientes; sin embargo, es común realizar las interconexiones mediante una red Ethernet dedicada de alta velocidad. Gracias a esta red de interconexión los nodos del clúster intercambian entre sí las tareas, las actualizaciones de estado y los datos del programa. En un clúster abierto, existirá una interfaz de red que conecte al clúster con el mundo exterior (internet).

Cuando se trata de resolver un problema en paralelo, el software debe ser capaz de dividirlo en tareas más pequeñas, repartirlas entre los nodos y elaborar los resultados. Puesto que las subtareas van a ejecutarse en paralelo se consigue un aumento de velocidad, aunque hay que tener en cuenta el retardo en la división, el reparto y la transmisión de mensajes (resultado, coherencia y estados).

En el caso de los clústeres de balanceo de carga, el hardware y el software deben actuar conjuntamente para que el tráfico se distribuya entre los nodos del clúster. De esta forma, se pueden ofrecer los servicios a mayor velocidad o se realiza una tarea más rápidamente.

Los servidores de un clúster de alta disponibilidad normalmente no comparten la carga de procesamiento que tiene un clúster de alto rendimiento; tampoco comparten la carga de tráfico, como lo hacen los clúster de balanceo de carga. Su función es la de estar preparados para entrar inmediatamente en funcionamiento, en caso de que falle algún otro servidor.

Software para la implementación de un clúster

En la actualidad hay varios tipos de software para la implementación y administración del clúster. A continuación se muestra un análisis comparativo de las mejores herramientas que se pueden implementar.

	Descripción	Alternativas
Software para clustering	Alta disponibilidad	Pirahna Open Mosix
	Balanceo de carga	Ultra Monkey Pirahna Open Mosix
	Configuración e instalación	FAI SIS System Image System Installer System Configuration
	Monitorización e Instalación	LVSmonn, Sincopt, Fsync, Ghosts, y Pconsole.
	Monitorización	Mon Heartbeat Fake Coda Ganglia

¿Por qué es necesaria la implementación de un clúster?

Ubicados en el ámbito informático y tecnológico, los usuarios tienen diversas necesidades en los diferentes escenarios (hogar, empresarial, educativo) y exigen satisfacerlas. Ello fomenta la importancia de la funcionalidad y la disponibilidad sin interrupción de los diversos servicios de internet, reproducción y transferencia de información de todo tipo, prestados por los recursos implicados. En el mercado existen soluciones varias, especializadas y sofisticadas, para brindar los servicios requeridos. Sus especificidades técnicas requieren ser concebidas con procesamiento, almacenamiento, redundancia, disponibilidad y desempeño de la forma más evolucionada, debido al desarrollo de la tecnología existente.

Tales soluciones tecnológicas requieren mucho dinero para su adquisición, lo que las posiciona en un grupo de mercado exclusivo. Aun así, cuando son adquiridas no se pueden separar de la depreciación, pues se convierten en modelos viejos y obsoletos. Esto dificulta en gran medida a la corporación que la adoptó, el actualizar o cambiar de tecnología por una solución equiparable o de mayor envergadura, que supla sus necesidades y cumpla su proyección a un bajo costo.

Por lo anterior, la solución de construir y configurar un clúster (independientemente del tipo que sea) para la reproducción de vídeo o transferencia de la información por internet, brindará a la entidad que lo adquiera servicios web más ágiles, escalabilidad y costos bajos en su desarrollo e implementación. Además, es una herramienta tecnológica que permitirá en forma adecuada atender las solicitudes web de los usuarios y transferir, simultáneamente, todo tipo de información por la internet, en forma rápida, íntegra, nítida e ininterrumpida. El clúster se ha construido de una manera minuciosa, lo cual garantiza factores óptimos de calidad como integridad, eficiencia y facilidad de mantenimiento, entre otros.

Las limitantes de sus capacidades de atención se limitan al crecimiento, ya que a mayor necesidad mayor debe ser la cantidad de recursos tecnológicos (nodos) disponibles en él. Esto no es un impedimento gracias a que puede adoptar diversas arquitecturas computacionales para su escalamiento. Su restricción real es el espacio físico disponible en el que se ubica la solución para albergar los nodos requeridos en pos de una atención eficaz a cada petición realizada por el usuario.

Construir un clúster puede aportar importantes ventajas en gran variedad de aplicaciones y ambientes. Por ejemplo, en las investigaciones meteorológicas y de pronóstico numérico del estado del tiempo, se requiere el manejo de cantidades masivas de datos y cálculos muy complejos. Al combinar el poder de muchas máquinas del tipo estación de trabajo o servidor se alcanzan niveles de rendimiento similares a los de las supercomputadoras, pero a menor costo. Otra situación de aplicabilidad de un clúster sería en un sitio web de mucho tráfico. Si no se cuenta con un plan de alta disponibilidad, cualquier problema menor en una tarjeta de red puede hacer que un servidor quede completamente inutilizado. Pero al contar con servidores redundantes y servidores de respaldo instantáneos, se puede reparar el problema mientras el sitio sigue funcionando sin suspensión de servicio.

En este proyecto se detalla una serie de definiciones el cual veremos a continuación:

1. Clúster:

El término clúster (del inglés cluster, "grupo" o "raíz") se aplica a los conjuntos o conglomerados de computadoras unidos entre sí normalmente por una red de alta velocidad y que se comportan como si fuesen una única computadora.

2. Alta disponibilidad:

Son clústeres cuyo objetivo de diseño es el de proveer disponibilidad y confiabilidad. Estos clústeres tratan de brindar la máxima disponibilidad de los servicios que ofrecen. La confiabilidad se provee mediante software que detecta fallos y permite recuperarse frente a los mismos, mientras que en hardware se evita tener un único punto de fallos.

3. Nodos:

es un punto de intersección, conexión o unión de varios elementos que confluyen en el mismo lugar, En redes de computadoras cada una de las máquinas es un nodo, y si la red es Internet, cada servidor constituye también un nodo.

Los clústers son usualmente empleados para mejorar el rendimiento y/o la disponibilidad por encima de la que es provista por un solo computador típicamente siendo más económico que computadores individuales de rapidez y disponibilidad comparables.

De un clúster se espera que presente combinaciones de los siguientes servicios:

- 1. Alto rendimiento**
- 2. Alta disponibilidad**
- 3. Balanceo de carga**
- 4. Escalabilidad**

Clasificación de los Clúster, según su configuración.

● **Homogéneos:**

Pueden tener todos la misma configuración de hardware y sistema operativo.

● **Semi homogéneos:**

Diferente rendimiento pero con arquitecturas y sistemas operativos similares.

● **Heterogéneos:**

Tienen diferente hardware y sistema operativo, lo que hace más fácil y económica su construcción.

Beneficios de la tecnología clúster:

Las aplicaciones paralelas escalables requieren: buen rendimiento, baja latencia, comunicaciones que dispongan de gran ancho de banda, redes escalables y acceso rápido a archivos. Un clúster puede satisfacer estos requisitos usando los recursos que tiene asociados a él; La tecnología clúster permite a las organizaciones incrementar su capacidad de procesamiento usando tecnología estándar, tanto en componentes de hardware como de software que pueden adquirirse a un costo relativamente bajo.

Apache

Apache es un poderoso servidor web, cuyo nombre proviene de la frase inglesa “a patchy server” y es completamente libre, ya que es un software Open Source y con licencia GPL. Una de las ventajas más grandes de Apache, es que es un servidor web multiplataforma, es decir, puede trabajar con diferentes sistemas operativos y mantener su excelente rendimiento.

Desde el año 1996, es el servidor web más popular del mundo, debido a su estabilidad y seguridad. Apache sigue siendo desarrollado por la comunidad de usuarios desarrolladores que trabaja bajo la tutela de Apache Software Foundation.

Principales características de Apache

Entre las principales características de Apache, se encuentran 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.

Características

Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos HTTP.

- **Multiplataforma.**
- **Modular:** Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- **Extensible:** gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

¿Qué es DRBD?

DRBD (Distributed Replicated Block Device) se refiere a los dispositivos de bloque concebido como una edificación en bloque, agrupados para formar un cluster de alta disponibilidad (HA). Esto se hace por medio de reflejo completo de un dispositivo de bloque asignado a través de una red.

Descripción del proyecto:

- Implementar un Clúster de Servidor de Ficheros de Alta Disponibilidad con heartbeat y DRBD.
- Se deberán configurar dos servidores de archivos que contengan la misma información y que se repliquen de manera que cuando un servidor falle, el servidor activo brinde el servicio sin interrupción del servicio.

Requisitos necesarios para la implementación del proyecto.

1. Tres computadoras (o máquinas virtuales), un cliente y dos servidores
2. Sistema Operativo GNU/Linux.
3. Debian stretch versión 9.0
4. Apache versión 2.4.25
5. Oracle VM VirtualBox versión 5.2
6. DRBD versión.

EL servidor de archivos brinda a los usuarios, poder almacenar de forma centraliza sus documentos, permitiendo una mejor experiencia en la web y una mejor; confianza de que puedan acceder a sus archivos en cualquier momento. El hecho de realizarlo de esta manera es que proporciona muchas ventajas en el entorno: EL usuario puede cambiar el dispositivo sin perder el acceso a los ficheros, el administrador de sistemas puede controlar correctamente el acceso a los mismos estableciendo los niveles de seguridad necesarios, controlar el espacio físico que ocupan los datos, hacer copias de seguridad.

Sección de términos

Términos	Significado
HA (alta disponibilidad)	(High Availability) es un sistema orientado a ofrecer y garantizar servicios en Alta Disponibilidad, es decir, con un alto grado de fiabilidad y de continuidad operativa.
Heartbeat	pulso o “latido” mediante el cual se mantiene la comunicación entre los nodos del clúster. Si el nodo activo no responde al latido, el nodo pasivo toma el control y despliega de inmediato los servicios replicados
Clúster	conjunto de computadoras construidas mediante la utilización de componentes de hardware que se comportan como si fuesen una única computadora.
NAS (Network Attached Storage)	es un dispositivo específico dedicado al almacenamiento a través de red
Balaneo de carga	es un sistema muy utilizado cuando se tiene que atender mucho tráfico en una aplicación o sitio web.
DRBD (Distributed Replicated Block Device)	es un software que permite hacer réplica de los datos de una partición entre varias máquinas.
Apache	es el servidor Web HTTP de código abierto
SSH	es un protocolo que sirve para conectarse remotamente a las máquinas de una red.
NFS (Network File System)	es un protocolo utilizado para compartir archivos entre las máquinas de una red.

Descripción del proyecto final.

Demostrar la funcionalidad de un clúster con servicio web en Apache y junto con la herramienta heartbeat y DRBD. El presente proyecto se desarrollo en tres nodos o maquinas virtuales los cuales se encarga de brindar alta disponibilidad. Para ellos tendremos dos servidores apache conectado a un balanceador de carga para que divida las solicitudes entrantes entre los nodos de apache. Sabemos que un balanceador de carga para los dos nodos de apache puede llegar a ser otro punto de falla, pues si este se cae el balanceador de carga las solicitudes no se repartirán a los nodos del servidor Apache. Como queremos proporcionar alta disponibilidad, El balanceador de carga, también se compondrá de dos nodos así si un Balanceador de carga falla, el otro responderá logrando mayor disponibilidad en el servidor.

La ventaja de utilizar un equilibrador de carga en comparación con el uso de DNS round robin según lo investigado es que el balanceador tratara de enviar las solicitudes al nodo con menos carga. Además de eso, si uno de los nodos de Apache se cae, el balanceador de carga se dará cuenta de eso y dirigirá todas las peticiones entrantes al nodo restante . Para esta configuración, necesitamos cuatro nodos (dos nodos Apache y dos nodos equilibrador de carga).

Diagrama de GANTT.

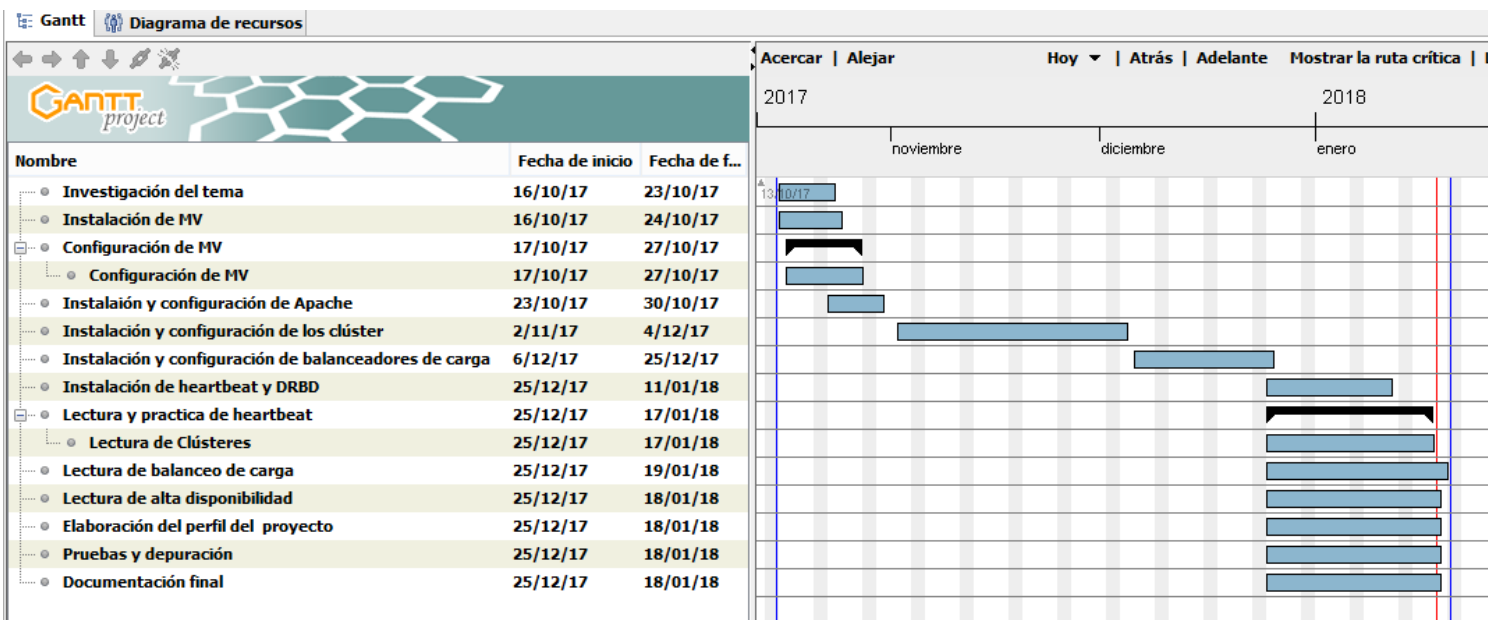
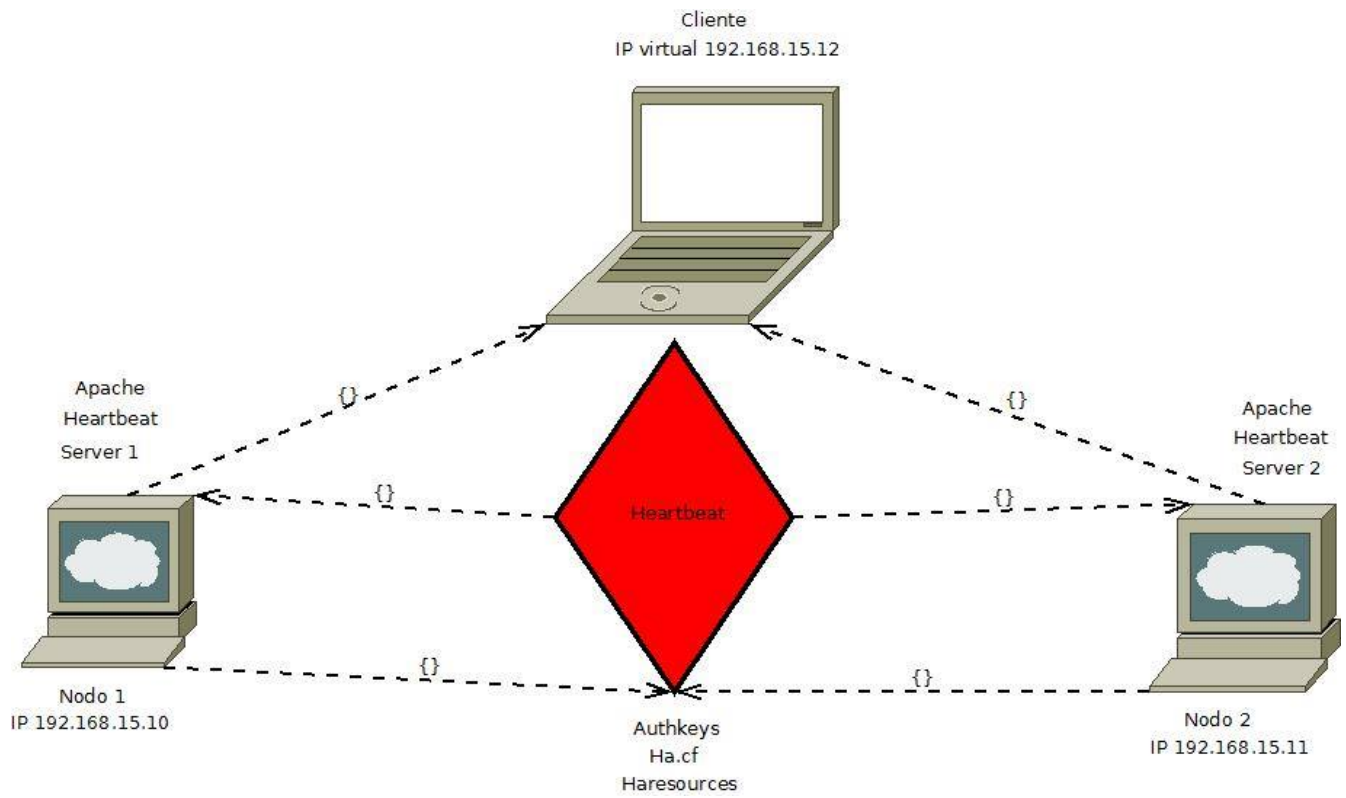


Diagrama de red.



Presupuesto.

Recursos materiales.

EL precio puede variar de acuerdo al hardware

Los costos para servidores Linux andan en un valor por mes de \$750.00

Dispositivo	característica	Precio
Una Laptop	Marca HP. Intel Pentium Processor N3540 (2.16GHz). 4GB DDR3L-1600 Memory. 500GB HDD Hard Drive	\$249,99

Sistema Operativo	Versión	Costo.
Debian	Para 64 bits	\$0.00

Viabilidad

En el presente sistema se desarrolla un sistema de alta disponibilidad, ya que debido a las nuevas tecnologías, la mediana y grandes empresas requieren hacer uso de ellas; para poder cumplir sus metas y objetivos ya sea a nivel social, económico y productividad. El cual consiste en llevar a cabo un clúster web de alta disponibilidad con tecnología Libre (GNU/LINUX), lo que implica ahorros en costos de licenciamiento y evita restricciones que las herramientas privativas imponen. Además al ser herramientas libres aseguramos que el conocimiento se pueda compartir y seguir desarrollando quizá por otras personas interesadas en ahondar mas en el tema.

Ventajas de un clúster web de alta disponibilidad.

- ◆ **Disponibilidad:** Capacidad para continuar operando ante la caída de alguno de los ordenadores del clúster.
- ◆ **Distribución en paralelo.**
- ◆ **Flexibilidad:** Los balanceadores de carga no están amarrados a ninguna arquitectura específica, en lo que respecta a hardware.
- ◆ **Costos:** El diseño y montaje requiere de inversiones sumamente bajas comparadas con las alternativas de solución, las cuales son de un costo elevado.
- ◆ **Escalabilidad:** Capacidad para hacer frente a volúmenes de trabajo cada vez mayores, prestando así un nivel de rendimiento óptimo.
- ◆ **Expansibilidad:** Capacidad de aumentar sus capacidades a través de mejores técnicas.
- ◆ Transferencia de información y todo tipo de servicio por internet de forma rápida, a bajo costo e ininterrumpidamente.
- ◆ Incremento de velocidad de procesamiento ofrecido por los clústeres de alto rendimiento.
- ◆ Incremento del número de transacciones o velocidad de respuesta ofrecido por los clústeres de balanceo de carga.
- ◆ Incremento de la confiabilidad y la robustez ofrecido por los clúster de alta disponibilidad.

Desventajas.

- ◆ Empresas y entidades prefieren seguir utilizando el modelo cliente/servidor tradicional debido al espacio físico o a nuevos problemas que no se daban en la arquitectura tradicional.

Desarrollo del prototipo.

Instalación de las maquinas virtuales

Herramientas necesarias.

1. ISO de instalación del sistema operativo Debian Stretch 9.0
2. Apache2 versión (2.4.25-3)
3. Heartbeat Versión (1:3.0.6-5)
4. Virtualbox versión 5.2.5
5. Switch que interconecte los equipos.
6. DRBD versión 9.0

Procedimiento de Instalación.

Instalación de virtual box.

1. En consola como root actualizamos los repositorios y los paquetes, e instalamos el paquete virtualbox y también virtualbox-guest-x11:
2. #su(contraseña)
3. #aptitude update
4. #aptitude upgrade
5. #aptitude install virtualbox virtualbox-guest-x11
6. Automaticamente instala las dependencias : dkms,virtualbox-dkms y virtualbox-qt.

Agregar nuestro usuario al grupo vboxuser

Tenemos que agregar nuestro usuario al grupo vboxusers para poder utilizar los dispositivos USB en las maquinas virtuales que vayamos a instalar. Para hacer esto, debemos abrir la terminal y ejecutar el siguiente comando como superusuario:

```
#adduser tu_usuario vboxusers
```

Para comprobar que el usuario se agrego correctamente al grupo, ejecutamos:

```
#adduser tu_usuario vboxusers
```

Y nos dirá “El usuario “nombre” ya es un miembro de 'vboxusers'.
Después reiniciamos el sistema.

Buenas prácticas para la construcción del proyecto

Una buena práctica puede ser una acción muy sencilla como también un conjunto de acciones de mayor complejidad y magnitud siempre que sean prácticas y buenas. Para la construcción del proyecto se han tomado en cuenta las siguientes.

Defina el Alcance y los Objetivos del Proyecto.

La creación de un servidor de ficheros de alta disponibilidad es el objetivo que se desea alcanzar con este proyecto, tomando en cuenta que sera a través de un cluster y utilizando maquinas virtuales las cuales serán creadas y configuradas con Oracle VM Virtualbox, Apache2, Heartbeat utilizando todos los recursos de software libre y hardware necesarios para ello.

El alcance o área de competencia define los límites del proyecto.

Cuando hablamos de buenas prácticas nos referimos a aquellas prácticas profesionales que resultan ser las mejores de entre todas las que los profesionales realizan para lograr los resultados esperados por sus clientes. Una buena práctica puede ser lo tanto una acción muy sencilla como también un conjunto de acciones de mayor complejidad y magnitud siempre que sean prácticas y buenas. Para la construcción de nuestro proyecto se han tomado en cuenta las siguientes.

Tareas:

1. Instalar en una maquina física Oracle VM Virtualbox.
2. Crear un clúster con maquinas virtuales, este estará constituido por dos nodos que serán los servidores mas un equipo cliente que puede ser físico o virtual.
3. Realizar la investigación de los paquetes de software requeridos para la realización del proyecto.
4. Instalación de los paquetes descargados.
5. configuración de maquinas y software necesario para el funcionamiento del clúster.
6. configuración de las IP en los nodos y cliente, así como la IP virtual que servirá de enlace entre los servidores para darles alternancia.
7. Realizar pruebas de funcionamiento del clúster
8. puesta en marcha y producción del clúster.

Conclusión.

La configuración de un clúster que funcione como un servidor de alta disponibilidad, nos ha permitido aprender sobre que software utilizar para lograrlo, y vencer los obstáculos que en el campo de las pruebas fueron apareciendo, así como solventar los errores que cometimos al crear los nodos virtuales y las configuraciones necesarias para lograr el objetivo.

Crear un clúster con dos nodos(servidores) y una maquina virtual como cliente, para ejecutar las pruebas así como realizar la investigación de que software eran necesarios para la realización del mismo, teniendo en cuenta la ayuda que nos brindara el docente. Nos ha permitido ejecutar el presente proyecto con éxito, no sin antes pasar por todos los problemas y trabajos emanados de las practicas realizadas, que nos dieron como resultado la experiencia de tener que usar software libre. Instalar Oracle VM Virtualbox, crear las maquinas virtuales, instalar Heartbeat, Apache2 sobre la plataforma de Debian, hacer las pruebas de configuración, solventar los errores que han aparecido asido gratificante pues nos ha permitido enriquecer nuestros conocimientos sobre que son, como funcionan, y para que sirve un clúster de alta disponibilidad.

Los clúster balanceadores de carga, alto desempeño y alta disponibilidad) es una área relativamente nueva en el mercado ya que a travez de esta tecnología se pueden realizar innovaciones. A pesar de existir clústeres construidos por grandes compañías y a un alto costo, todavía se pueden construir y configurar clústeres más accesibles en cuanto a su licencia y funcionalidad. Este es el primer paso para su diseño, construcción y configuración, con el fin de satisfacer requerimientos más específicos planteados por los administradores de clústeres.

Por otra parte el software libre ha permitido desarrollar todas las aplicaciones necesarias para construir el clúster web, demostrando su viabilidad y madurez para implementar soluciones que respondan a problemas reales.

Antes de este proyecto no sabíamos que era un servidor de ficheros de alta disponibilidad, ni como funcionaba, ahora podemos decir que hemos adquirido suficiente conocimiento como para configurar e instalar todo lo necesario para crearlo.

Recomendaciones.

- ◆ Hoy en día, a nivel empresarial es indispensable poner en practica las nuevas tecnologías informáticas. Ya que atravez de la implementación de este proyecto. La visualización y el desarrollo de los clúster web.
- ◆ Decidir si se va configurar en maquinas reales o en maquinas virtuales cada una de estas opciones tiene sus pro y sus contras.
- ◆ Descargar el software requerido recomendamos licencias de tipo: Licencia Pública General de GNU (GPL) o una licencia propietaria con características adicionales.
- ◆ Si se realizara con maquinas virtuales ver si la maquina física puede aceptar virtualización. La computadora en la que se harán las maquinas virtuales deberá estar bien dotada de recursos de hardware capas de soportar la virtualización y llenar requerimientos tales como suficiente cantidad de memoria RAM, un buen Disco Duro, y un microprocesador (CPU) con una buena velocidad, para que a la hora de arrancar las maquinas virtuales no se ponga lenta.

Anexos.

Instalación y preparación de MV (maquinas virtuales)

Herramientas necesarias

1. ISO de instalación del sistema operativo Debian stretch 9
2. Apache2 Versión (2.4.25-3)
3. Hearbeat Versión (1:3.0.5-5)
4. Virtualbox versión (5.2.5)
5. DRBD Versión

Vamos a montar 2 maquinas virtuales, con nombre pc01 y pc02, en las cuales estará instalado todo lo necesario para formar, el closter con heartbeat y apache2.

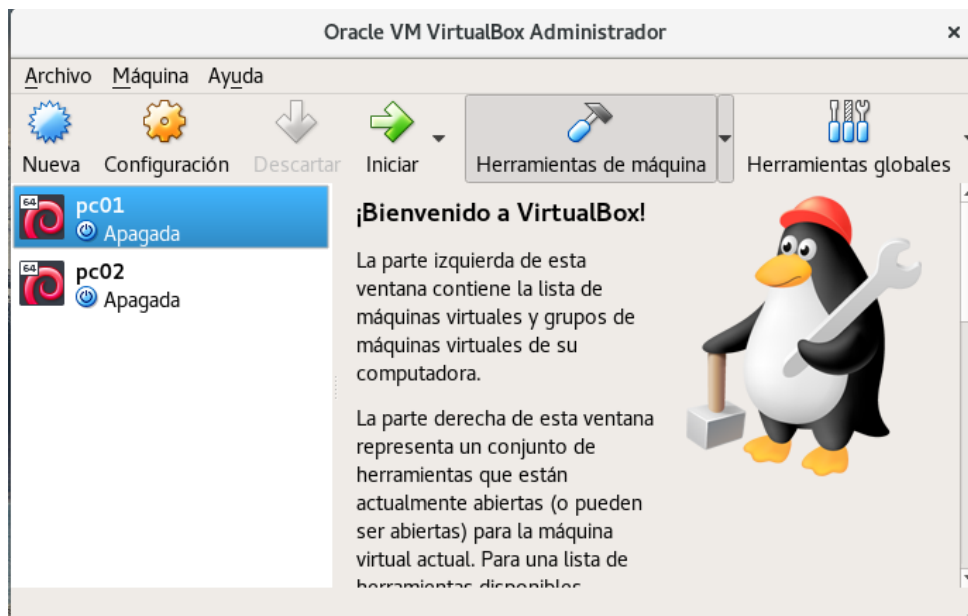


Figura : Creación de maquinas virtuales, pc01 y pc02.

Ahora configuraremos las interfaces de res de cada maquina virtual ;en adaptador puente para poder tener acceso a la actualización de los repositorios y la descarga de los paquetes a utilizar.

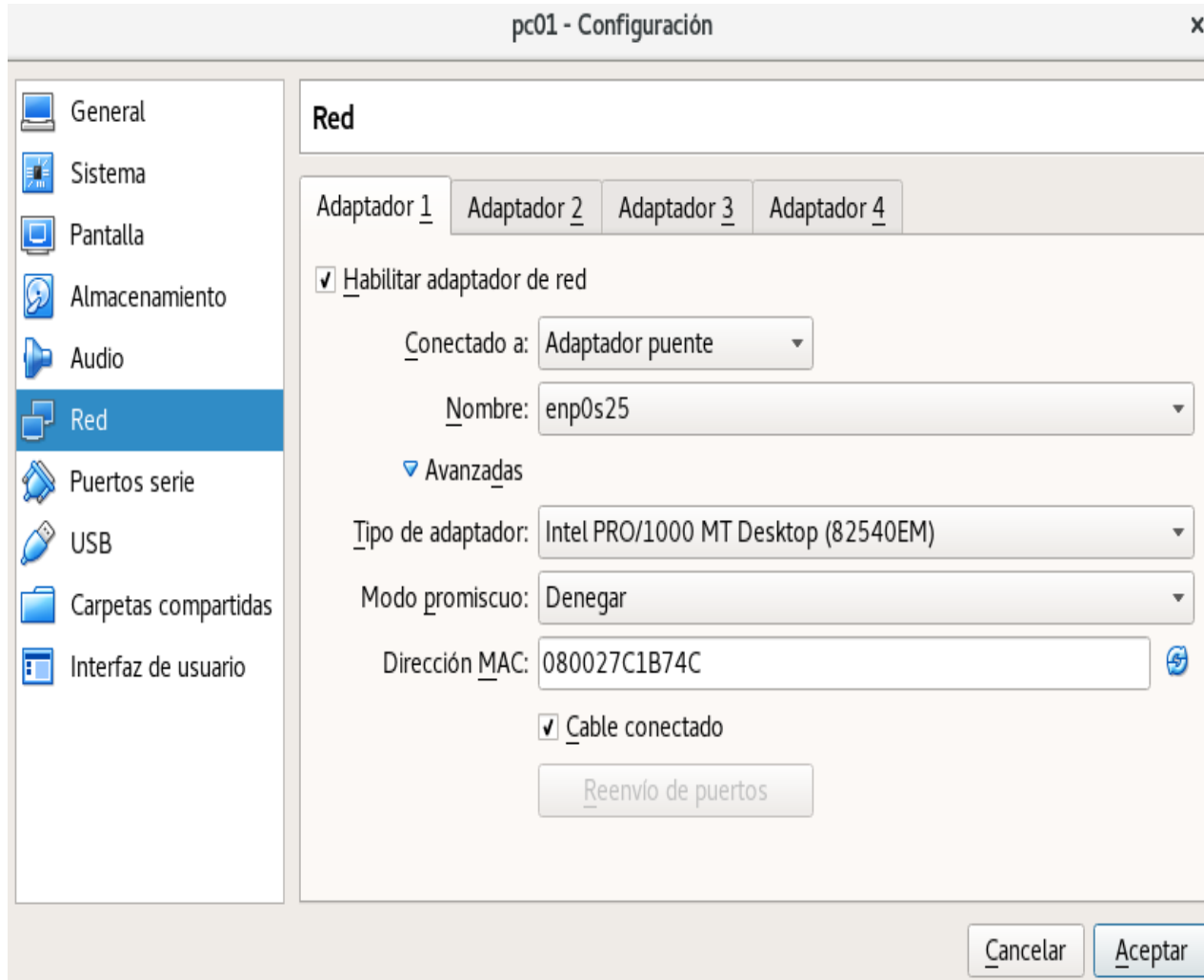


Figura: Configuración de red pc01

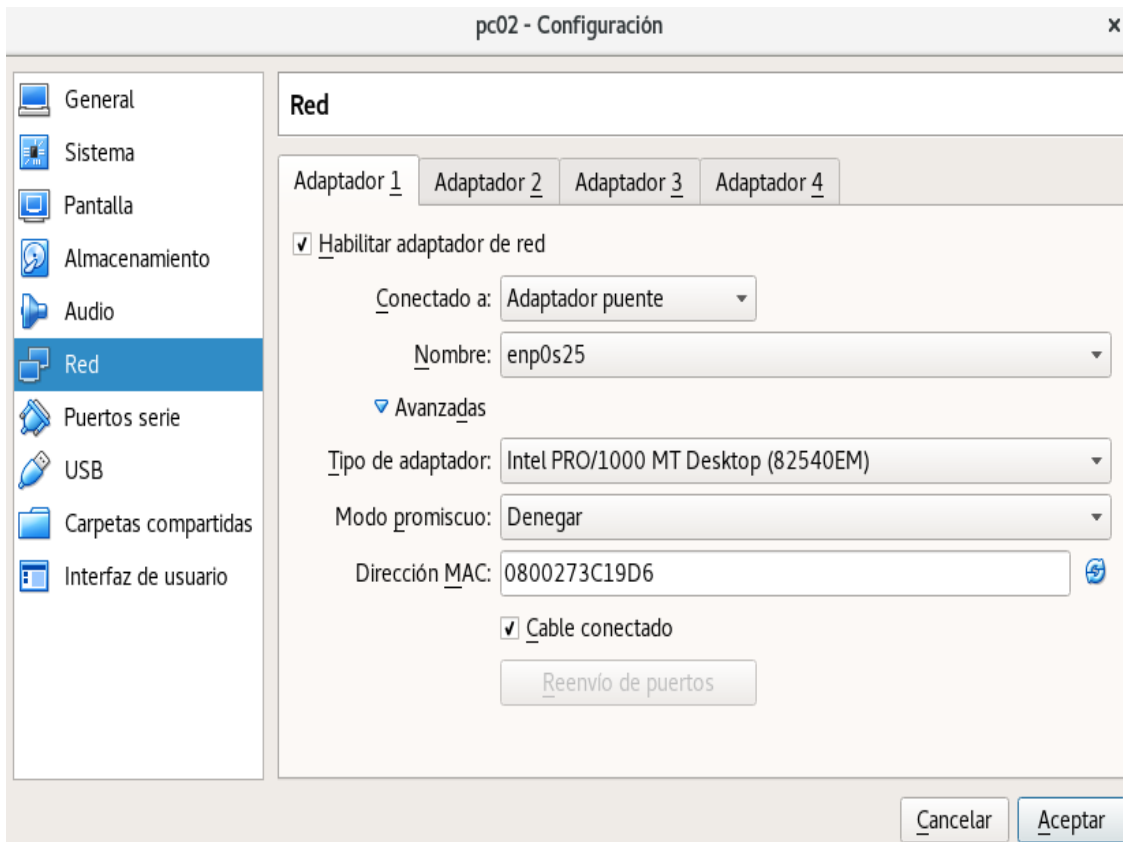


Figura: configuración de red pc02

Iniciamos las maquinas virtuales y nos lo guiamos como root y actualizamos nuestro repositorios. **#apt-get update** en las dos maquinas virtuales.

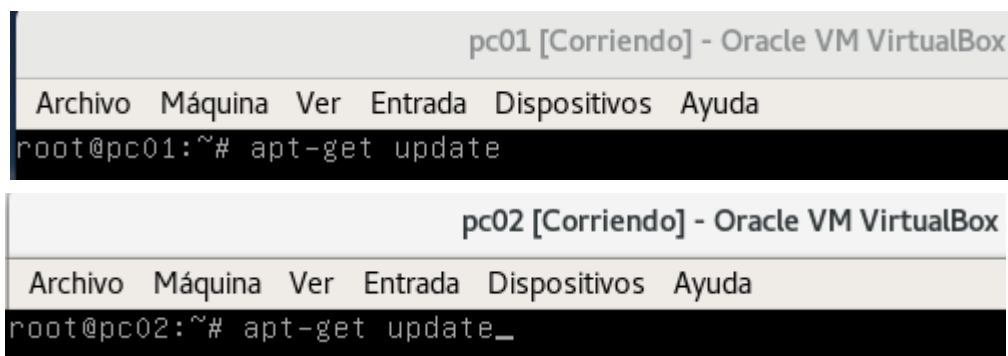


Figura : inicialización de MV y actualización de repositorios.

Instalación de Apache en MV (maquinas virtuales).

Ahora procedemos a instalar apache2

apt-get install apache2

Esto lo aremos de la misma manera en las dos maquinas virtuales.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@pc01:/home/pc1# apt-get install apache2
```

Luego detenemos el servicio

service apache2 stop

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@pc01:/home/pc1# service apache2 stop
```

Para evitar que el servicio apache2 se levante al inicio, ejecutamos el siguiente comando.

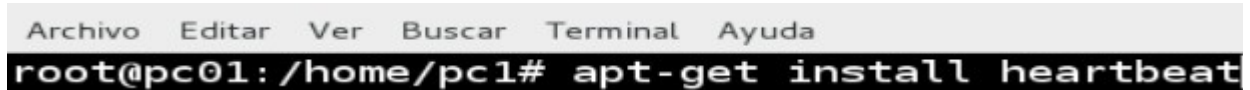
update-rc.d -f apache2 disable

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@pc01:/home/pc1# update-rc.d -f apache2 disable
```


Instalación de heartbeat.

A continuación instalamos HEARTBEAT:

```
# apt-get install heartbeat
```



```
Archivo Editar Ver Buscar Terminal Ayuda  
root@pc01:/home/pc1# apt-get install heartbeat
```

Tras la instalación de HEARTBEAT habrá aparecido en la ruta `/usr/share/doc/heartbeat`, la cual contiene 3 ficheros que debemos configurar para que HEARTBEAT funcione.

Estos 3 ficheros son:

authkeys: Fichero de autenticación entre nodos.

ha.cf: Fichero principal de configuración de HEARTBEAT.

Haresources: Fichero donde se indica cuál sera el nodo activo. Así como los servicios a gestionar.

Luego de haber instalado HEARTBEAT descomprimos y copiamos los ficheros de configuración de HEARTBEAT en la ruta `/etc/ha.d`

Descomprimos **ha.cf.gz**

```
#gzip -d /usr/share/doc/heartbeat/ha.cf.gz
```



```
Archivo Editar Ver Buscar Terminal Ayuda  
root@pc01:/home/pc1# gzip -d /usr/share/doc/heartbeat/ha.cf.gz
```

Ahora copiamos el fichero authkeys

```
#cp /usr/share/doc/heartbeat/authkeys /etc/ha.d/
```

Archivo Editar Ver Buscar Terminal Ayuda

```
root@pc01:/home/pc1# cp /usr/share/doc/heartbeat/authkeys /etc/ha.d/
```

Ahora copiamos el fichero ha.cf

```
#cp /usr/share/doc/heartbeat/ha.cf /etc/ha.d/
```

Archivo Editar Ver Buscar Terminal Ayuda

```
root@pc01:/home/pc1# cp /usr/share/doc/heartbeat/ha.cf /etc/ha.d/
```

Ahora copiamos el fichero haresource

```
#cp /usr/share/doc/heartbeat/haresource /etc/ha.d/
```

Archivo Editar Ver Buscar Terminal Ayuda

```
root@pc01:/home/pc1# cp /usr/share/doc/heartbeat/haresources /etc/ha.d/
```

Configuración de heartbeat.

Ahora empezamos a configurar cada uno de los tres archivos que copiamos, estas configuraciones serán igual para las dos maquinas virtuales exceptuando la Ip's que utilizaran para comunicarse y detectar cuando el nodo principal esta detenido.

authkeys: en este fichero estableceremos la contraseña de autenticación compartida por los dos nodos.
#nano /etc/ha.d/authkeys quedara de esta manera.

```
# 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!
```

Le damos permisos para que solo pueda ser accedido por el propietario del mismo
#chmod 600 /etc/ha.d/authkeys

Esta configuración sera igual en los dos nodos.

ha.cf: este fichero principal de configuración de HEARTBEAT. Aquí des comentaremos algunas lineas que son esenciales para el correcto funcionamiento de HEARTBEAT.

#nano /etc/ha.d/ha.cf

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

#Tiempo transcurrido hasta declarar al nodo como caído (segundos)
deadtime 5

Si durante 5 segundo no responde el otro nodo, este nodo está en alerta
wartime 5

```

#
keepalive 2
#
#       deadtime: how long-to-declare-host-dead
#
#       If you set this too low you will
#       split-brain (or cluster partition)
#       See the FAQ for how to use warn
#
deadtime 5
#
#       warntime: how long before issuing "late
#       See the FAQ for how to use warntime to
#
warntime 5
#
#

```

#Tiempo transcurrido para comenzar a levantar los servicios (segundos)

initdead 5

#Puerto para la comunicación UDP

udpport 694

```

#
initdead 5
#
#       What UDP port to use for bcast/ucast communication?
#
udpport 694

```

esta configuración sera igual en los dos nodos

#Interfaz utilizado para enviar los mensajes de broadcast

bcast enp0s3

```

bcast    enp0s3          # Linux
#bcast   eth1 eth2       # Linux
#bcast   le0             # Solaris
#bcast   le1 le2        # Solaris
#
#       Set up a multicast heartbeat medium
#       mcast [dev] [mcast group] [port] [ttl] [loop]

```

#interfaz y nodo al que enviar los latidos

uicast enp0s3

```
[peer-ip-addr] IP address of peer to send packets
#
uicast enp0s3 192.168.15.11
#
[peer-ip-addr] IP address of peer to send packets
#
uicast enp0s3 192.168.15.10
#
```

La Ip que colocaremos sera la Ip del nodo contrario.

Si el nodo1 tiene Ip 192.168.15.10 entonces la configuración cruzada sera **192.168.15.11**

Si el nodo2 tiene Ip 192.168.15.11 entonces la configuración cruzada sera **192.168.15.10**

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

Declaración de los dos nodos

esta configuración sera igual en los dos nodos

```
node      nodename ...
node      pc01
node      pc02
#
```

Haresource: este fichero contiene información de los recursos que deseamos que tenga alta disponibilidad. Al principio de este archivo estableceremos quien será el servidor que predominará. La IP virtual y el servicio que se brindará.

#nano /etc/ha.d/haresource

```
GNU nano 2.7.4      Fichero: /etc/ha.d/haresources
pc01 IPaddr::192.168.15.12/24/enp0s3 apache2
#
```

```
#nano /etc/ha.d/haresource
```

luego modificamos el fichero index.html ubicado en /var/www/html/index.html

```
#nano /var/www/html/index.html
```

para tener una prueba gráfica del servicio, esto lo aremos en las dos maquinas virtuales.

```
pc01 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@pc01:~# nano /var/www/html/index.html _
```

```
pc02 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@pc02:~# nano /var/www/html/index.html _
```

```
pc01 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
GNU nano 2.7.4 Fichero: /var/www/html/index.html
<DOCTYPE html!>
<html>
  <head>
    <title> hola pc01 </title>
  </head>
  <body>
    <h1> computadora pc01 </h1>
  </body>
</html>
```

```
pc02 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
GNU nano 2.7.4 Fichero: /var/www/html/index.html
<DOCTYPE html!>
<html>
  <head>
    <title> hola pc02 </title>
  </head>
  <body>
    <h1> computadora pc02 </h1>
  </body>
</html>
```

Reiniciamos el servicio de HEARTBEAT en ambos nodos

```
/etc/init.d/heartbeat stop
/etc/init.d/heartbeat start
```

luego encendemos las maquinas y levantamos el servicio

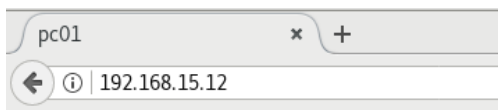
```
service heartbeat start
```

En ambas maquinas y listo.

Luego en la pc01 detenemos el servicio

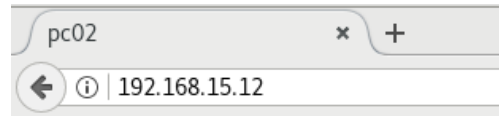
```
/etc/init.d/heartbeat stop
```

Al realizar esto automáticamente nos levantara el servicio desde la pc02 sin cambiar ninguna Ip asumiendo que el servidor sigue funcionando sin ningún problema.



pc01

192.168.15.10



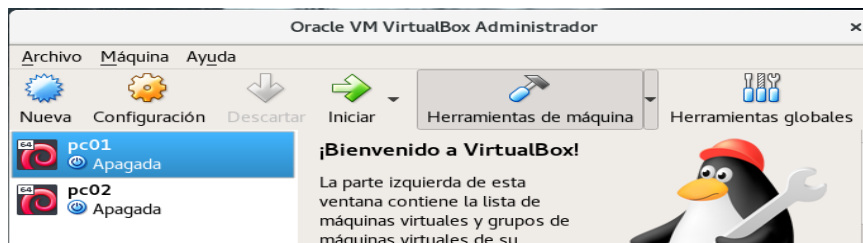
pc02

192.168.15.11

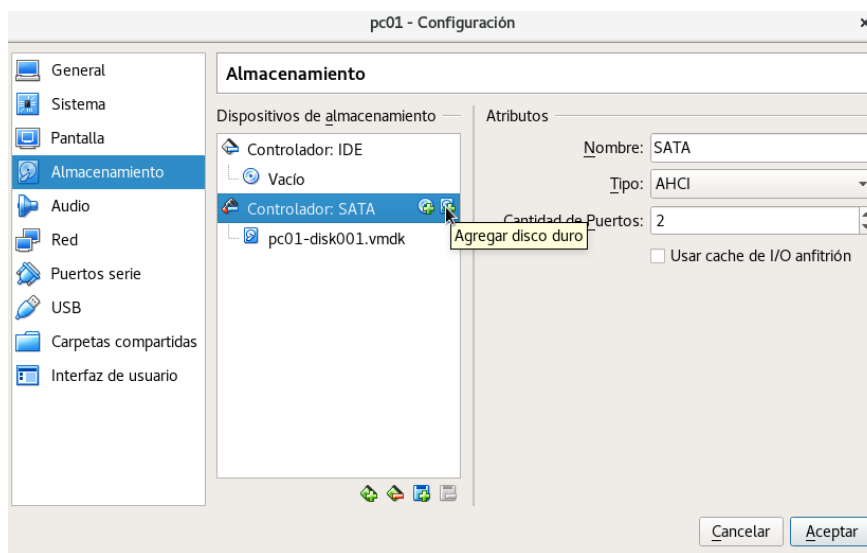
Agregar una unidad de almacenamiento

Este proceso sera igual en las dos computadoras virtuales, ya que DRBD requiere que las unidades de almacenamiento que se utilizaran para montar el directorio, sean de las misma capacidad de almacenamiento.

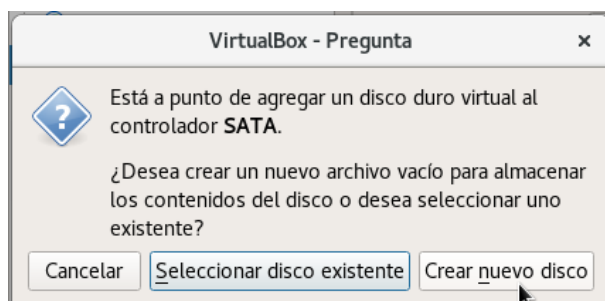
Paso N.º 1: Seleccionamos la maquina virtual y dar click en configuraciones.



Paso N.º 2: Damos click en el icono de almacenamiento, nos vamos donde dice Controlador. SATA, damos click en agregar disco duro.



Paso N.º 3: Nos aparecerá una ventana como esta, le damos click el botón Crear nuevo disco.



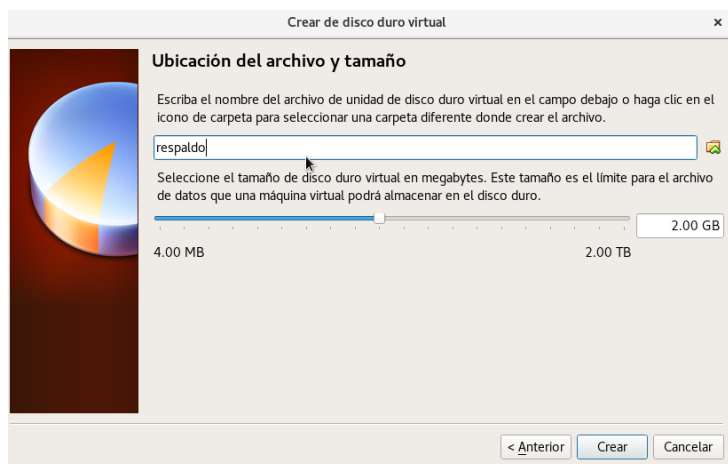
Paso N.º 4: dejamos seleccionada la opción VDI, solo damos click en siguiente.



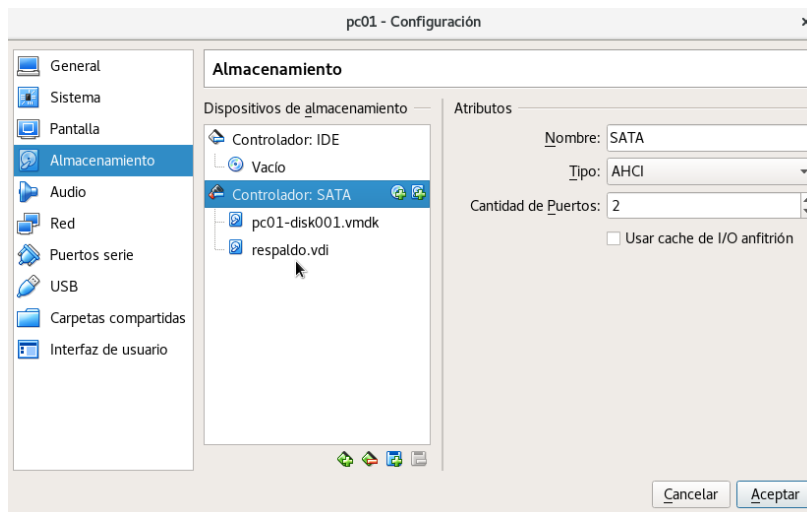
Paso N.º 5: Dejamos seleccionada la opción Reservado dinámicamente, damos click en el botón siguiente.



Paso N.º 6: Colocamos un nombre a la nueva unidad de almacenamiento, asignamos la cantidad de espacio en GB que tendrá la unidad y luego damos click en el botón Crear.



Paso N.º 7: Como vemos en la imagen, ya hemos creado una nueva unidad de almacenamiento virtual con las opciones que nos ofrece VirtualBox, ahora solo le damos click en el botón aceptar.



Formateo de la nueva unidad de almacenamiento

Esto sera igual en los dos maquinas virtuales.

Paso N.º 1: Esto ya en la terminal como usuario root, usamos el comando fdisk y con el argumento -l para que nos liste las unidades de almacenamiento que tenemos.

```
pc01 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@pc01:~# fdisk -l
Disco /dev/sda: 10 GiB, 10737418240 bytes, 20971520 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: dos
Identificador del disco: 0x223db065

Disposit.  Inicio Comienzo   Final Sectores Tamaño Id Tipo
/dev/sda1 *      2048 18874367 18872320    9G 83 Linux
/dev/sda2      18876414 20969471 2093058   1022M  5 Extendida
/dev/sda5      18876416 20969471 2093056   1022M  82 Linux swap / Solaris

Disco /dev/sdb: 2 GiB, 2147483648 bytes, 4194304 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
root@pc01:~# _
```

Como podemos ver, el sistema reconoce la nueva unidad de almacenamiento que hemos creado.

Paso N.º 2: Pondremos esta ruta /dev/sdb utilizando la herramienta fdisk

```
pc01 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@pc01:~# fdisk /dev/sdb_
```

Paso N.º 3: tecleamos n. damos un enter

```
pc01 [Corriendo] - Oracle VM VirtualBox x
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@pc01:~# fdisk /dev/sdb
Bienvenido a fdisk (util-linux 2.29.2).
Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.
Tenga cuidado antes de utilizar la orden de escritura.

El dispositivo no contiene una tabla de particiones reconocida.
Se ha creado una nueva etiqueta de disco DOS con el identificador de disco 0xbff
f7be70.

Orden (m para obtener ayuda): n
```

Paso N.º 4: Tecleamos p. click en enter

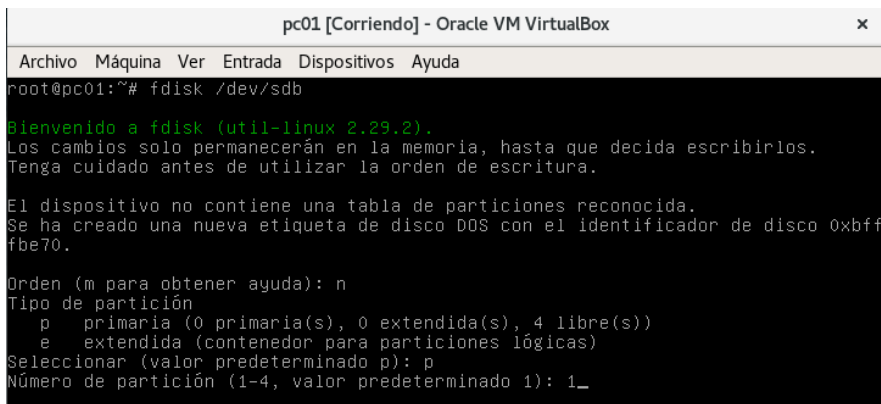
```
pc01 [Corriendo] - Oracle VM VirtualBox x
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@pc01:~# fdisk /dev/sdb
Bienvenido a fdisk (util-linux 2.29.2).
Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.
Tenga cuidado antes de utilizar la orden de escritura.

El dispositivo no contiene una tabla de particiones reconocida.
Se ha creado una nueva etiqueta de disco DOS con el identificador de disco 0xbff
f7be70.

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

Paso N.º 5: Número de partición (1-4)

Teclamos 1. click en enter



```
pc01 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@pc01:~# fdisk /dev/sdb
Bienvenido a fdisk (util-linux 2.29.2).
Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.
Tenga cuidado antes de utilizar la orden de escritura.

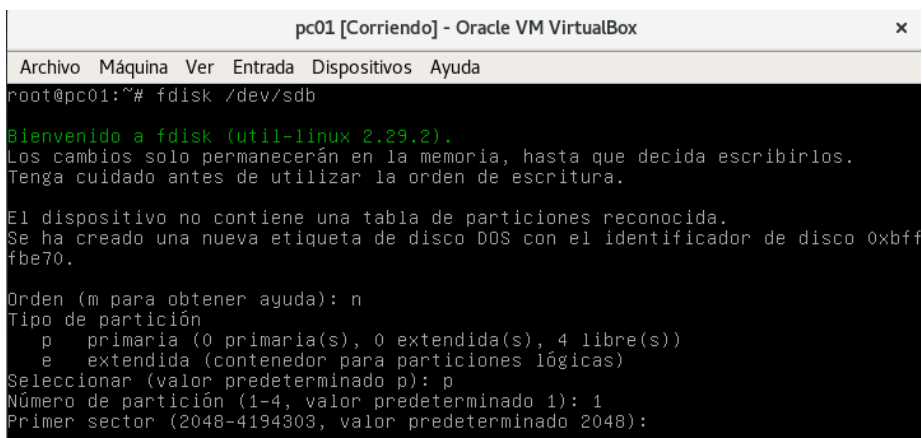
El dispositivo no contiene una tabla de particiones reconocida.
Se ha creado una nueva etiqueta de disco DOS con el identificador de disco 0xbff
f7be70.

Orden (m para obtener ayuda): n
Tipo de partición
  p  primaria (0 primaria(s), 0 extendida(s), 4 libre(s))
  e  extendida (contenedor para particiones lógicas)
Seleccionar (valor predeterminado p): p
Número de partición (1-4, valor predeterminado 1): 1_
```

Paso N.º 6: Primer cilindro (1-48641, valor predeterminado 1):

Teclamos Enter.

Se está utilizando el valor predeterminado 1



```
pc01 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@pc01:~# fdisk /dev/sdb
Bienvenido a fdisk (util-linux 2.29.2).
Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.
Tenga cuidado antes de utilizar la orden de escritura.

El dispositivo no contiene una tabla de particiones reconocida.
Se ha creado una nueva etiqueta de disco DOS con el identificador de disco 0xbff
f7be70.

Orden (m para obtener ayuda): n
Tipo de partición
  p  primaria (0 primaria(s), 0 extendida(s), 4 libre(s))
  e  extendida (contenedor para particiones lógicas)
Seleccionar (valor predeterminado p): p
Número de partición (1-4, valor predeterminado 1): 1
Primer sector (2048-4194303, valor predeterminado 2048):
```

Paso N.º 7: Último cilindro o +tamaño o +tamañoM o +tamañoK (1-48641, valor predeterminado 48641):

Tecleamos **Enter**

```
pc01 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@pc01:~# fdisk /dev/sdb
Bienvenido a fdisk (util-linux 2.29.2).
Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.
Tenga cuidado antes de utilizar la orden de escritura.

El dispositivo no contiene una tabla de particiones reconocida.
Se ha creado una nueva etiqueta de disco DOS con el identificador de disco 0xbff
f7e70.

Orden (m para obtener ayuda): n
Tipo de partición
  p  primaria (0 primaria(s), 0 extendida(s), 4 libre(s))
  e  extendida (contenedor para particiones lógicas)
Seleccionar (valor predeterminado p): p
Número de partición (1-4, valor predeterminado 1): 1
Primer sector (2048-4194303, valor predeterminado 2048):
Último sector, +sectores o +tamaño{K,M,G,T,P} (2048-4194303, valor predeterminad
o 4194303): 1_
```

Paso N.º 8: Tecleamos: **w** click en enter

¡Se ha modificado la tabla de particiones!

```
root@pc01:~# fdisk /dev/sdb
Bienvenido a fdisk (util-linux 2.29.2).
Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.
Tenga cuidado antes de utilizar la orden de escritura.

El dispositivo no contiene una tabla de particiones reconocida.
Se ha creado una nueva etiqueta de disco DOS con el identificador de disco 0xc84
8f2e6.

Orden (m para obtener ayuda): n
Tipo de partición
  p  primaria (0 primaria(s), 0 extendida(s), 4 libre(s))
  e  extendida (contenedor para particiones lógicas)
Seleccionar (valor predeterminado p): p
Número de partición (1-4, valor predeterminado 1): 1
Primer sector (2048-4194303, valor predeterminado 2048):
Último sector, +sectores o +tamaño{K,M,G,T,P} (2048-4194303, valor predeterminad
o 4194303):
Crea una nueva partición 1 de tipo 'Linux' y de tamaño 2 GiB.

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

root@pc01:~#
```

Paso N.º 9: Se habrá creado el disco sdb1. Le asignamos el sistema de archivos ext4 tecleando lo siguiente.

mkfs.ext4 /dev/sdb1

```
root@pc01:~# mkfs.ext4 /dev/sdb1
mke2fs 1.43.4 (31-Jan-2017)
Se está creando un sistema de ficheros con 524032 bloques de 4k y 131072 nodos-1
UUID del sistema de ficheros: 7933ca16-2081-44be-b079-a97a563a8bec
Respaldo del superbloque guardado en los bloques:
    32768, 98304, 163840, 229376, 294912

Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Creando el fichero de transacciones (8192 bloques): hecho
Escribiendo superbloques y la información contable del sistema de ficheros: 0/1
hecho
root@pc01:~#
```

Instalación de drbd

Paso N.º 1: Instalaremos drbd en las dos maquinas virtuales usando la siguiente linea de comandos.

apt-get install drbd-utils

```
root@pc01:~# apt-get install drbd-utils
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  heirloom-mailx s-nail
Paquetes sugeridos:
  heartbeat exim4 | mail-transport-agent
Se instalarán los siguientes paquetes NUEVOS:
  drbd-utils heirloom-mailx s-nail
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 881 kB de archivos.
Se utilizarán 2,469 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] _
```

Paso N.º 2: configuraremos las interfaces de red, de forma estática, usando el editor de texto nano, colocando esta ruta /etc/network/interfaces, en ambos equipos pc01 y pc02

nano /etc/network/interfaces

```
GNU nano 2.7.4 Fichero: /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
#allow-hotplug enp0s3
#iface enp0s3 inet dhcp

auto enp0s3
iface enp0s3 inet static
    address 192.168.15.10
    netmask 255.255.255.0
    network 192.168.15.0
    broadcast 192.168.15.255
    gateway 192.168.15.1
```

```
GNU nano 2.7.4 Fichero: /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
#allow-hotplug enp0s3
#iface enp0s3 inet dhcp

auto enp0s3
iface enp0s3 inet static
    address 192.168.15.11
    netmask 255.255.255.0
    network 192.168.15.0
    broadcast 192.168.15.255
    gateway 192.168.15.1
```

Quedara configuradas de esta manera.

Paso N.º 3: creamos el archivo de configuración en esta ruta /etc/drbd.d/, le colocamos un nombre como por ejemplo r1.res, siempre tiene que llevar la extensión *.res, El contenido de este fichero es idéntico en ambos servidores.

```
nano /etc/drbd.d/r1.res
```

```
Archivo Máquina Ver Entrada Dispositivos /
root@pc01:~# nano /etc/drbd.d/r1.res
```

```
resource r1{
protocol C;
startup {
    wfc-timeout 15;
    degr-wfc-timeout 60;
}

meta-disk internal;
device /dev/drbd0;
syncer {
    verify-alg sha1;
}
net {
    allow-two-primaries;
}
on pc01 {
    disk /dev/sdb1;
    address 192.168.15.10:7789;
}
on pc02 {
    disk /dev/sdb1;
    address 192.168.15.11:7789;
}
```

Dentro del archivo colocamos lo siguiente:
En este archivo configuraremos el nombre de la unidad virtual de almacenamiento donde creamos un sistema de fichero ext4, los nombres de los servidores con sus Ips estáticas y el puerto a utilizar.

Paso N.º 4: En ambos servidores creamos el dispositivo drbd para el recurso r1 mediante el siguiente comando:

```
drbdadm create-md r1
```

```
Archivo Máquina Ver Entrada Dispositivos /
root@pc01:~# drbdadm create-md r1

initializing activity log
NOT initializing bitmap
Writing meta data...
New drbd meta data block successfully created.
root@pc01:~# _
```

Paso N.º 5: Los siguientes dos pasos, solo es necesario realizarlos en uno de los dos servidores. En el ejemplo lo ejecutamos en server1(pc01).
Establecemos el servidor como primary:

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

```
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@pc01:~# drbdadm -- --overwrite-data-of-peer primary all
```

Paso N.º 6: Ahora iniciamos el servicio en los dos servidores:

```
service drbd start
```

```
Archivo Máquina Ver Entrada Disposit  
root@pc01:~# service drbd start
```

Paso N.º 7: Formateamos el dispositivo drbd:

```
mkfs.ext4 /dev/drbd0
```

```
root@pc01:~# mkfs.ext4 /dev/drbd0
```

Paso N.º 8: Ahora creamos una carpeta llamada data, esto sera igual en ambos servidores, por ejemplo.

```
mkdir /data
```

```
Archivo Máquina Ver Entrada Disposit  
root@pc01:~# mkdir /data_
```

Paso N.º 9: Asignamos al server1 (pc01) que sera primary

```
drbdadm primary r1
```

```
Archivo Máquina Ver Entrada Dispositivo  
root@pc01:~# drbdadm primary r1
```

Paso N.º 10: Asignamos al server2 (pc02) que sera secondary

```
drbdadm secondary r1
```

```
Archivo Máquina Ver Entrada Dispositivo  
root@pc02:~# drbdadm secondary r1
```

Podemos ver como se van sincronizando los discos, esto es en ambos servidores

```
cat /proc/drbd
```


Servidor 1 (pc01)

```
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@pc01:~# cat /proc/drbd
version: 8.4.7 (api:1/proto:86-101)
srcversion: 0904DF2CCF7283ACE07D07A
0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
   ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
root@pc01:~#
```

servidor 2 (pc02)

```
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@pc02:~# cat /proc/drbd
version: 8.4.7 (api:1/proto:86-101)
srcversion: 0904DF2CCF7283ACE07D07A
0: cs:Connected ro:Secondary/Primary ds:UpToDate/UpToDate C r-----
   ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
root@pc02:~# _
```

Paso N.º11: Montamos la carpeta en la siguiente ruta /dev/drbd0

```
mount /dev/drbd0 /data
```

```
Archivo Máquina Ver Entrada Dispositivos Ay
root@pc01:~# mount /dev/drbd0 /data/
```

Luego veremos de forma listada los dispositivos montado en nuestro servidor1 (pc01)

```
df -h
```

```
root@pc01:~# df -h
S.ficheros      Tamaño Usados  Disp Uso% Montado en
udev            488M    0      488M  0% /dev
tmpfs           100M    3.0M    97M   3% /run
/dev/sda1       8.8G    1.3G    7.1G  16% /
tmpfs           499M    54M    445M  11% /dev/shm
tmpfs           5.0M    0       5.0M  0% /run/lock
tmpfs           499M    0       499M  0% /sys/fs/cgroup
tmpfs           100M    0       100M  0% /run/user/0
/dev/drbd0      2.0G    3.1M    1.9G  1% /data
root@pc01:~# _
```

Paso N.º 12: Creamos un archivo de texto en el fichero que hemos montado.
nano /data/prueba.txt

```
Archivo Máquina Ver Entrada Dispositivos Ay
root@pc01:~# nano /data/prueba.txt
```

Paso N.º 13: Montamos la unidad en el servidor 1(pc01), verificamos el fichero
ls -l /data

```
root@pc01:~# ls -l /data/
total 4
-rw-r--r-- 1 root root 6 ene 19 17:32 prueba.txt
root@pc01:~#
```

Automatización de los servicios de heartbeat y drbd

Paso N.º 1: Creamos un archivo .sh en la ruta /home/uls/, llamado h.sh, en ambos servidores, con la sintaxis siguiente.

nano /home/uls/h.sh

```
GNU nano 2.7.4
#!/bin/sh
service heartbeat start
service drbd start
```

Paso N°2: En el archivo de interfaces de red, agregaremos la siguiente línea de comando post-up
/home/uls/h.sh

```
GNU nano 2.7.4 Fichero: /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
#allow-hotplug enp0s3
#iface enp0s3 inet dhcp

auto enp0s3
iface enp0s3 inet static
    address 192.168.15.10
    netmask 255.255.255.0

post-up /home/uls/h.sh
```

Este comando sirve para iniciar los servicios de heartbeat y drbd des pues que nuestro equipo inicie y cargue las interfaces de red.

Paso N°3: En el archivo /etc/ha.d/haresources agregamos la configuración de montaje de forma automática, al reiniciar nuestros equipos de la partición y el directorio que hemos creado previamente llamado /data, esta configuración sera igual en los dos nodos.

```
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.7.4 Fichero: /etc/ha.d/haresources
pc01 IPaddr::192.168.15.12/24/enp0s3 drbddisk::r1 Filesystem::/dev/drbd0::/data::ext4 apache2
#
# 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
```

Bibliografía.

Consultado el 26 de octubre del 2017

Revista Digital Sociedad de la Información. Año Junio 2008.

<http://www.sociedadelainformacion.com/13/cluster.pdf>

Consultado el 26 de octubre del 2017

Revista Digital Sociedad de la Información

<http://www.sociedadelainformacion.com/13/cluster.pdf>

Consultado el 25 de octubre del 2017

Utilización y Administración avanzadas de sistemas GNU/Linux y aplicaciones Software

Libre para estudiantes universitarios. Autor Angel de Bustos

Pérez<jadebustos@augcyl.org>.

Consultado el 28 de octubre del 2017

Utilización y Administración avanzadas de sistemas GNU/Linux y aplicaciones Software

Libre para estudiantes universitarios. Por José Angel de Bustos

Pérez<jadebustos@augcyl.org>.

Consultado 3 de enero del 2018

Publicado el 14 Febrero 2014 Escrito por Javier Orovengua

<https://www.linux-party.com/99-cloudcomputing/9045-configurar-un-servidor-raid1-en-red-con-drbd-en-linux-2-de-2>

Consultado el 1 de enero del 2018.

Wikipedia. Título “Cluster Beowulf”

https://es.wikipedia.org/wiki/Cluster_Beowulf

Consultado el 1 de enero del 2018.

Wikipedia. Título “Clúster (informática)”

[https://es.wikipedia.org/wiki/Cl%C3%BAster_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Cl%C3%BAster_(inform%C3%A1tica))

Consultado el 7 de enero de 2018

Publicado el 14 Febrero 2014 Escrito por Javier Orovengua

<https://www.linux-party.com/99-cloudcomputing/9045-configurar-un-servidor-raid1-en-red-con-drbd-en-linux-2-de-2>