UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

IMPLEMENTACIÓN DE CONTROLES DE SEGURIDAD DE LA INFORMACIÓN MEDIANTE SOFTWARE OPEN SOURCE.

CAMPOVERDE TOLEDO EDGAR ANDREY
INGENIERO DE SISTEMAS

MACHALA
2019

UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

IMPLEMENTACIÓN DE CONTROLES DE SEGURIDAD DE LA INFORMACIÓN MEDIANTE SOFTWARE OPEN SOURCE.

CAMPOVERDE TOLEDO EDGAR ANDREY
INGENIERO DE SISTEMAS

MACHALA
2019

UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

EXAMEN COMPLEXIVO

IMPLEMENTACIÓN DE CONTROLES DE SEGURIDAD DE LA INFORMACIÓN MEDIANTE SOFTWARE OPEN SOURCE.

CAMPOVERDE TOLEDO EDGAR ANDREY
INGENIERO DE SISTEMAS

RIVAS ASANZA WILMER BRAULIO

MACHALA, 22 DE AGOSTO DE 2019

MACHALA
22 de agosto de 2019

Nota de aceptación:

Quienes suscriben, en nuestra condición de evaluadores del trabajo de titulación denominado IMPLEMENTACIÓN DE CONTROLES DE SEGURIDAD DE LA INFORMACIÓN MEDIANTE SOFTWARE OPEN SOURCE., hacemos constar que luego de haber revisado el manuscrito del precitado trabajo, consideramos que reúne las condiciones académicas para continuar con la fase de evaluación correspondiente.

RIVAS ASANZA WILMER BRAULIO
0702580192
TUTOR - ESPECIALISTA 1

VALAREZO PARDO MILTON RAFAEL
0704518893
ESPECIALISTA 2

REDROVAN CASTILLO FAUSTO FABIAN
0702739228
ESPECIALISTA 3

Fecha de impresión: jueves 22 de agosto de 2019 - 14:16

URKUND

Urkund Analysis Result

Analysed Document: informe_casoPractico Campoverde.docx (D54802022)
Submitted: 8/13/2019 4:00:00 PM
Submitted By: wrivas@utmachala.edu.ec
Significance: 0 %

Sources included in the report:

Instances where selected sources appear:

0

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

El que suscribe, CAMPOVERDE TOLEDO EDGAR ANDREY, en calidad de autor del siguiente trabajo escrito titulado IMPLEMENTACIÓN DE CONTROLES DE SEGURIDAD DE LA INFORMACIÓN MEDIANTE SOFTWARE OPEN SOURCE., otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

El autor declara que el contenido que se publicará es de carácter académico y se enmarca en las dispociones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

El autor como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 22 de agosto de 2019

CAMPOVERDE TOLEDO EDGAR ANDREY
0704496660

DEDICATORIA

Quiero dedicar este trabajo únicamente a mis padres, agradecerles por todo el apoyo brindado en los momentos más difíciles y por haberme dado la motivación necesaria para culminar con éxito mis estudios.

RESUMEN

IMPLEMENTACIÓN DE CONTROLES DE SEGURIDAD DE LA INFORMACIÓN MEDIANTE SOFTWARE OPEN SOURCE

En la actualidad cualquier tipo de negocio depende de las tecnologías de la información para poder realizar sus procesos, estas tecnologías generan datos, estos datos luego de ser organizados y analizados se convierten en información, esta información puede tener un alto valor. Por lo cual, se requiere la aplicación de ciertos controles que nos permitan cumplir con los principios fundamentales de la seguridad de la información para poder mantener la integridad, disponibilidad y confidencialidad.

El presente trabajo exhibe la implementación de ciertos controles de seguridad que nos ayudarán a mantener la información íntegra y siempre disponible. Para ello, por sus características de código abierto, se utilizó el sistema operativo CentOS 7, basado en la distribución Red Hat Enterprise Linux. Mediante el cual, con la ayuda de herramientas como rsync y rsnapshot, se realizó una configuración de servidores espejo entre dos nodos, cada uno con los mismos servicios, efectuando así la ejecución de respaldos de seguridad automatizados y la sincronización de carpetas específicas. Además, se ejecutó la replicación de un sitio web, a través de la implementación de un clúster conformado por ambos nodos o servidores.

Como resultado, se logró cumplir con el objetivo de la mitigación de ciertos riesgos de seguridad de la información del negocio, manteniendo la disponibilidad de la información considerada más importante por la empresa, en varios sitios en tiempo real y la continuidad del sitio web frente a fallos de cualquier tipo.

Palabras claves:

Seguridad, Controles, Clúster, Nodo, Servidor Espejo, Rsync, Rsnapshot, CentOS.

ABSTRACT

IMPLEMENTATION OF INFORMATION SECURITY CONTROLS THROUGH OPEN SOURCE SOFTWARE

Nowadays, any type of business depends on the information technologies to be able to carry out its processes, these technologies generate data, these data after being organized and analyzed become information, this information can have a high value. Therefore, the application of certain controls that allow us to comply with the fundamental principles of information security is required in order to maintain integrity, availability and confidentiality.

This work shows the implementation of certain security controls that will help us keep the information complete and always available. For this, due to its open source features, the CentOS 7 operating system was used, based on the Red Hat Enterprise Linux distribution. Through which, with the help of tools such as rsync and rsnapshot, a mirror server configuration was made between two nodes, each with the same services, thus performing the execution of automated security backups and the synchronization of specific folders. In addition, the replication of a website was executed, through the implementation of a cluster consisting of both nodes or servers.

As a result, the objective of mitigating certain business information security risks was achieved, maintaining the availability of the information considered most important by the company, in several real-time sites and the continuity of the website against failures of any kind.

Key words:

Security, Controls, Cluster, Node, Mirror Server, Rsync, Rsnapshot, CentOS.

CONTENIDO

pág.

LISTA DE ILUSTRACIONES

pág.

LISTA DE TABLAS

pág.

INTRODUCCIÓN

La información es el activo más valioso que poseen las organizaciones, por lo cual su integridad y confidencialidad son factores muy importantes a tener en cuenta, ya que una mala administración o pérdida de la misma, sea ocasionada por el ser humano, problemas de hardware o algún evento de catástrofes naturales, generaría grandes pérdidas económicas y sobre todo mal prestigio para el negocio.

La seguridad de la información se compone de diferentes aspectos como: seguridad de acceso y dispositivos, manipulación de contraseñas, control de vulnerabilidades y más. Todos estos requieren un estudio, un presupuesto y una ejecución, ya sea preventiva o correctiva. [1]

Hoy en día gracias a la innovación de la tecnología, mantener siempre activo algún servicio y a salvo los datos de una empresa es más fácil de lo que parece. Existen varios métodos y herramientas que nos ayudan con procedimientos de alta disponibilidad y replicación de datos como medida de seguridad preventiva para mitigar riesgos de continuidad.

Para esto se ha creado una solución llamada servidores espejo (mirrors), que según López en [2] se basan en conservar una copia de la información contenida en uno o varios de otros servidores. De tal manera, si por alguna razón el servidor principal se cae, el tráfico de red se desvía al servidor espejo y los servicios seguirán habilitados.

Este trabajo fue desarrollado en el sistema operativo CentOS 7 de tipo open source, el mismo que Georgopoulou en [3] define como una guía alternativa de desarrollo y distribución de software basada en los principios del libre intercambio de información y la contribución abierta. Detalla la implementación de un clúster entre dos servidores espejo y la réplica de un servicio web, la replicación de un servidor se utiliza para configurar dos servidores y sincronizar sus datos, de tal manera que ambos contengan una copia exacta del mismo volumen. [4] También se configuró la sincronización de carpetas y respaldos automáticos de archivos de tipo incremental. Constituye de una introducción, planteamiento del problema a resolver y un desarrollo compuesto por el marco teórico junto con todas las características y elaboración, más los resultados tras haber cumplido con los objetivos.

Problema General

Con el avance de las tecnologías de la información y comunicación (TICS), los datos de una empresa han adquirido un gran valor, sobre todo comercial. Por lo tanto, se torna una necesidad implementar controles de seguridad de la información, ya que al no hacerlo todos estos datos no estarán seguros, se verán comprometidos, expuestos y vulnerables frente a cualquier tipo de amenazas o catástrofes naturales.

Pero, se ha encontrado un problema muy común al momento de implementar dichas soluciones; esto se debe al costo de adquisición de un software de pago o mediante servicio privado y el mismo en ocasiones no satisface por completo nuestras necesidades.

Objetivo General

Implementar controles de seguridad de la información a través del uso de herramientas tipo software libre para mitigar los riesgos de continuidad del negocio.

Objetivos Específicos

- Configurar un clúster y replicar un servicio web mediante una dirección IP virtual.
- Configurar servidores espejo a nivel de archivos.
- Realizar respaldos automatizados de archivos y carpetas.

1. DESARROLLO

1.1 Marco Teórico

En este marco teórico se da a conocer acerca de la seguridad de la información, ciertos controles que se pueden tomar al respecto para conservar los datos seguros, implementaciones de seguridad con software libre para la continuidad de servicios de red y la importancia de la alta disponibilidad. Se define en [5] como software libre aquel que luego de ser adquirido, puede ser utilizado, copiado, analizado, modificado y comercializado libremente.

1.1.1 *Seguridad de la información.* La seguridad de la información básicamente busca preservar la confidencialidad, integridad y disponibilidad de los datos más importantes de una organización. Para cumplir este objetivo, se han creado ciertas normas o estándares internacionales de seguridad. En la tabla 1 se puede observar un esquema de controles según cada uno de los principios básicos de la seguridad de la información.

Tabla 1 Esquema de controles de la seguridad de la información

Esquema de controles		
Integridad	**Confidencialidad**	**Disponibilidad**
Pública	*Nominal*	*Recuperable*
Restringida	*Estándar*	*Manual*
Confidencial	*Individual*	*Automática*
Secreta	*Doble Intervención*	*Inmediata*

Fuente: Elaboración Propia

Un estándar para la seguridad de la información reside en un grupo de reglas, las mismas que tienen como propósito normalizar las operaciones en una empresa, aplicando un énfasis en la gestión y aseguramiento de la información. [6] La gestión de la seguridad de la información está detallada por el estándar ISO/IEC 27001, este estándar internacional ha sido preparado para proporcionar requisitos para diseñar, implementar, conservar y mejorar continuamente un Sistema de Gestión de Seguridad de la Información (SGSI). [7] Dichos sistemas están encaminados a la identificación e implementación de seguridad para reducir el riesgo en el manejo de la información. [8]

1.1.2 *Gestión de la seguridad de la información.* El enfoque de la selección de los controles de seguridad de la información, también es un rol muy importante a tener en cuenta, no depende únicamente de la cantidad de controles que se puedan implementar para conseguir un buen nivel de seguridad, sino también la competencia del conjunto de controles. [9]

La gestión de la seguridad de la información se alcanza a través de la implementación de un conjunto de controles que abarcan políticas, procedimientos, estructuras organizadas y sistemas de software y hardware. [10]

1.1.3 *Automatización de controles de seguridad de la información.* El termino general de automatización, implica la ejecución o acción autónoma, sin intervención o requerimiento de actuación humana. La automatización de los controles de la seguridad de la información, se encargan de operar, monitorizar y revisar la misma, de manera autónoma, utilizando software informáticos, herramientas de hardware; sin necesidad de la intervención de acciones humanas. [10]

La gestión automática de controles de seguridad de la información, es una de las potenciales rutas para conseguir que la gestión de la seguridad de la información sea un asunto menos complicado y más seguro en un ambiente de frecuentes amenazas de seguridad, tomando en cuenta la abundancia de controles a implementar. [11]

1.1.4 *Ciberseguridad.* De acuerdo a la Asociación de Auditoría y Control de Sistemas de Información (ISACA) la ciberseguridad es “La protección de los activos de información de amenazas dirigidas a la información procesada, almacenada y transportada por sistemas de información interconectados”

La seguridad cibernética y la seguridad de la información son mencionadas indistintamente, pero la seguridad cibernética es un componente de la seguridad de la información. [12]

1.1.5 *Servidores espejo.* Un servidor espejo, es un servidor secundario el cual posee la característica de contener exactamente la misma información y realizar las funciones de otro servidor. Es recomendable el uso de servidores espejo, ya que reducen el tiempo de conexión e impiden la sobrecarga de los servidores principales. [13]

En sistemas en los cuales es de suma importancia conservar la disponibilidad de los datos, se aprueba la configuración de un servidor espejo, garantizando así la minimización de la reparación del sistema ante imprevistos. [14]

El uso más común de servidores espejo es en empresas que ofrecen servicios a través de su sitio web con constante tráfico. Cuando un servidor deja de funcionar ya sea por problemas de red, fallos de energía o hardware, el espejo con la información duplicada, asume su función de principal y entra en línea permitiendo llevar a cabo la continuidad del negocio.

La principal funcionalidad de implementar este servicio es mantener siempre el tráfico y que no se evidencie la caída del mismo. El artículo [15] menciona que, el tráfico es una variable que varía su conducta según el servicio (contenido web, correo electrónico, sistemas de monitoreo y alarma, etc.).

1.1.6 *Clúster y alta disponibilidad.* Michael Porter define clúster como “un grupo de compañías y asociaciones interconectadas, las cuales están geográficamente cerca, se desempeñan en un sector de industria similar, y están unidas por una serie de características comunes y complementarias”. [16]

En el ambiente de la informática, un clúster está representado por un grupo de computadoras independientes, interconectadas entre si, que actúan como si fueran una sola. Como resultado, se consigue aumentar el rendimiento al trabajar varias unidades juntas y añadir sus potenciales. [17]

Una ventaja de un clúster es que no es obligatorio que los dispositivos que lo integran sean iguales a nivel de hardware, e incluso que tengan el mismo sistema operativo.

Existen varios tipos de clúster clasificados según su función, los cuales son:

1.1.6.1 Balanceo de carga. Diseñado para compartir alto tráfico entre varios servidores y equilibrar las cargas de trabajo entre todos los nodos.

1.1.6.2 *Alto rendimiento.* Ejecutan procesos o tareas con necesidades de grandes cantidades de memoria de forma simultanea o en paralelo, lo cual ayuda a mejorar rendimiento de aplicaciones.

1.1.6.3 *Alta disponibilidad.* La redundancia es la base de la alta disponibilidad, pues al haber varios nodos redundantes, en caso de haber alguna falla en el nodo principal, el servicio se traslada desde el nodo con falla a otro nodo del clúster completamente activo y así continuar ofreciendo el servicio. Un clúster de alta disponibilidad son conjuntos de varios servidores que trabajan de manera simultánea, compartiendo datos y servicios tales como el servidor web Apache, el gestor de bases de datos MySQL, sistemas de ficheros y otros. [17], [18]

1.1.7 *Sistemas de alta disponibilidad.* Para mitigar los problemas de la pérdida de servicios a causa de eventos imprevistos, surge la implementación de los sistemas de alta disponibilidad. Los sistemas de alta disponibilidad lo conforman elementos de hardware y software que, integrados y configurados para funcionar en conjunto, facilitan el acceso continuo a los datos y aplicaciones. [18], [17]

1.1.8 *Respaldos de seguridad.* Un respaldo o backup es una copia adicional de la información de cualquier dispositivo, la cual sirve para posteriormente hacer una restauración de los archivos en otro dispositivo secundario o en el mismo.

Los equipos de almacenamiento masivo de información están propensos a tener fallas, ahí es donde reside la importancia de contar con una copia de seguridad de la información. [19]

La frecuencia de un respaldo se verá definida con base en la periodicidad de los cambios de la información. En la tabla 2 se detalla los diferentes tipos de respaldo que existen.

Tabla 2 Tipos de Respaldos

Tipos de Respaldos	
Completo	*Es una copia total de todos los datos del sistema a respaldar.*
Incremental	*Primero se necesita tener un respaldo completo, partiendo de este, solo se copian los datos modificados desde el último respaldo de cualquier tipo.*
Diferencial	*Se necesita partir de un respaldo completo, posterior a eso, se puede realizar el backup de los archivos modificados desde el ultimo respaldo completo.*

Fuente: Elaboración Propia

1.1.8.1 Copia de seguridad en espejo. Una copia de seguridad en espejo se basa en realizar un respaldo exacto de todos los datos. Estas copias se pueden hacer en "caliente"; es decir, mientras se esté trabajando con los datos originales, se hace una copia espejo en otro disco alterno. La ventaja de usar estas copias es que no contienen archivos antiguos o inservibles, pero también esto puede resultar un problema, ya que, si se borra un archivo por accidente en el servidor original, en el espejo será de igual manera.

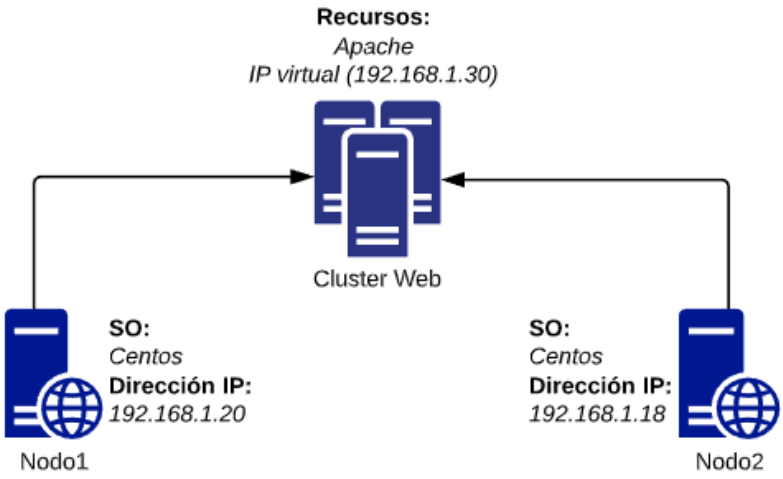
1.1.8.2 Sincronización de directorios. En el área de la gestión de servidores, realizar respaldos de manera automatizada y precisa conlleva a una alta disponibilidad de servicios y continuidad del negocio. La sincronización de directorios entre servidores es un proceso automatizado que asegura tener en tiempo real las mismas versiones de directorios y el mismo contenido en múltiples ubicaciones.

1.2 Resultados

Para dejar evidenciado del desarrollo de este trabajo, se tuvo que realizar la instalación y configuración de dos servidores con CentOS 7. Ambos servidores tienen implementado el servicio web de Apache con la misma página web y la herramienta rsync, pero cada uno asignado sus respectivas direcciones IP.

Se creó un clúster conformado por 2 servidores como nodos, se creó una dirección IP virtual para representar los nodos del clúster, la misma que se le asignó un recurso de tipo web para que el cliente pueda acceder al web server a través de ella. La figura 1 muestra la representación general del clúster.

Figura 1 Representación de clúster



Fuente: Elaboración propia

El nodo 1 es el servidor principal en el cual por defecto se ejecuta el servicio web y la IP virtual. Pero, cuando el nodo 1 presenta alguna falla y no responde, el nodo 2 pasa a cumplir esa función de principal replicando el servicio para mantener la disponibilidad.

1.2.1 *Configuración de clúster.* Para la implementación del clúster, se utilizó los paquetes de corosync como componente de mensajería y comunicación, pacemaker como administrador de recursos y pcs como el gestor del clúster para facilitar la administración del mismo desde cualquier nodo, en la figura 2 se puede observar el estado final del clúster ya configurado. (Ver Anexo A)

Figura 2 Verificación de estado final del clúster

```
[root@nodo1 ~]# pcs cluster status
Cluster Status:
 Stack: corosync
 Current DC: nodo2 (version 1.1.19-8.el7_6.4-c3c624ea3d) - partition with quorum
 Last updated: Mon Jul 22 22:12:25 2019
 Last change: Wed Jul 17 16:03:14 2019 by root via cibadmin on nodo1
 2 nodes configured
 2 resources configured

PCSD Status:
  nodo1: Online
  nodo2: Online
```

Fuente: Elaboración propia

1.2.2 *Configuración dirección IP virtual (VIP).* Para cumplir con la necesidad de la replicación web, es obligatorio agregar y configurar una dirección IP virtual al clúster de tipo recurso, esta dirección VIP es la que se contactará para poder tener acceso al servicio web. (Ver Anexo B)

En la figura 3 se puede observar una comunicación exitosa con la dirección IP virtual.

Figura 3 Prueba de comunicación con dirección IP virtual

```
[root@nodo1 ~]# ping -c1 192.168.1.30
PING 192.168.1.30 (192.168.1.30) 56(84) bytes of data.
64 bytes from 192.168.1.30: icmp_seq=1 ttl=64 time=0.026 ms

--- 192.168.1.30 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.026/0.026/0.026/0.000 ms
```

Fuente: Elaboración propia

1.2.3 *Configuración de servidor web Apache.* Para replicar el servicio web mediante el clúster, se debe instalar el servicio de Apache y configurar la misma página web de manera estática en ambos nodos. (Ver Anexo C)

1.2.4 Configuración de servidor Rsync. Rsync es una herramienta de actualización remota que permite sincronizar un árbol de directorios locales con un servidor remoto. [20] Es una manera muy sencilla de realizar copias de respaldo tipo incremental de manera eficiente, rápida y segura. (Ver Anexo D).

Figura 4 Copia de archivos mediante rsync a servidor remoto

```
[root@nodo1 /]# rsync -avz /home/admin/Documentos/ respaldo@192.168.1.18:/home/r
espaldo/Escritorio
The authenticity of host '192.168.1.18 (192.168.1.18)' can't be established.
ECDSA key fingerprint is SHA256:SHK52z0GX/muhuIW1KhxhJ3rjCf5MZjq47bkD6D5a+8.
ECDSA key fingerprint is MD5:e4:02:a0:e9:8e:25:2d:4f:bc:28:05:71:f8:df:2e:c1.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.18' (ECDSA) to the list of known hosts.
respaldo@192.168.1.18's password:
sending incremental file list
./
prueba.txt

sent 135 bytes  received 38 bytes  7.36 bytes/sec
total size is 11  speedup is 0.06
```

Fuente: Elaboración propia

Existen dos opciones para sincronizar directorios de manera eficaz con rsync, ya sea mediante el cron o creando scripts. (Ver Anexo E)

1.2.5 *Configuración de respaldos automáticos.* Para configurar los respaldos automatizados de la información más importante para la empresa, se utilizó la herramienta rsnapshot basada en rsync. Rsnapshot permite crear copias incrementales de distintas maneras. Siendo: horarias, diarias, semanales, mensuales y anuales. (Ver Anexo F)

Figura 5 Prueba de configuración de archivo /etc/rsnapshot.conf

```
[root@nodo1 ~]# sudo rsnapshot configtest
Syntax OK
```

Fuente: Elaboración propia

2. CONCLUSIONES

- Se realizó exitosamente la implementación de los controles de seguridad de la información mediante el uso de herramientas de software libre, logrando mitigar los riesgos de continuidad.

- La configuración de los servidores espejo se pudo realizar de manera sencilla con la herramienta rsync, logrando copiar los archivos del servidor principal al secundario.

- El proceso de replicación del sitio web resultó algo tedioso, pero se logró cumplir con el objetivo mediante la implementación del clúster utilizando como recurso una dirección IP virtual asignada al servicio de Apache para que funcione con el mismo.

- Se logró realizar la configuración de respaldos automatizados con la ayuda de la herramienta rsnapshot, la misma que permite configurar los respaldos conforme a las necesidades de la empresa.

BIBLIOGRAFÍA

[1] J. Alberto, M. F. Andrés, and A. D. Fernando, “Estudio y gestión de vulnerabilidades informáticas para una empresa privada en el departamento de Boyacá (Colombia) Information Vulnerabilities ’ Study and Management , for a Private Enterprise in the Boyacá Colombian Department Estudo e gestão de vulne,” *M.Sc. Univ. St. Tomás (Tunja-Boyacá, Colomb. Fac. Ing.*, vol. 23, no. 37, pp. 65–72, 2014.

[2] D. R. Lopez, *Internet. la Red Con Mayusculas.* 1997.

[3] P. Georgopoulou, “The free / open source software movement Resistance or change ?,” 2009.

[4] J. P. Moreno Alvarado, “IMPLEMENTACIÓN DE UN SERVIDOR DE DOMINIO ESPEJO, BASADO EN EL ANALISIS Y DIAGNOSTICO DE FUNCIONAMIENTO DE LA RED LAN, EN LA FUNDACIÓN UNIVERSITARIA DE CIENCIAS DE LA SALUD EN LA CIUDAD DE BOGOTÁ,” 2018.

[5] D. García, A. María, and O. Cuello, “La promoción del uso del software libre por parte de las universidades Universities promotion of free software use,” 2007.

[6] M. Diéguez and C. Cares, “Comparación de dos enfoques cuantitativos para seleccionar controles de seguridad de la información,” pp. 113–128, 2019.

[7] ISO/IEC, “Information technology - Security techniques - Information security management systems - Overview and vocabulary,” *Iso/Iec*, vol. 2009, p. ISO/IEC 27000:2009(E), 2009.

[8] E. J. Favaloro, “Information Security Management Best Practice Based on ISO/IEC 17799,” *Semin. Thromb. Hemost.*, vol. 37, no. 8, pp. 863–867, 2011.

[9] A. Otero, A. Ejnioui, C. E. Otero, and G. Tejay, “Evaluation of Information Security Controls in Organizations by Grey Relational Analysis,” *Int. J. Dependable Trust. Inf. Syst.*, vol. 2, pp. 36–54, Sep. 2013.

[10] “Gestión automatizada e integrada de controles de seguridad informática,” *Rev. Ing. Electrónica, Automática y Comun.*, vol. 34, no. 1, pp. 40–58, 2013.

[11] M. Miranda Cairo, O. Valdés Puga, I. Pérez Mallea, and R. Portelles Cobas, Renier Sánchez Zequeira, “Metodología para la Implementación de la Gestión Automatizada de Controles de Seguridad Informática,” *Rev. Cuba. Ciencias Informáticas*, vol. 10, no. 2, pp. 14–26, 2016.

[12] R. Sabillon, “A Practical Model to Perform Comprehensive Cybersecurity Audits,” *Enfoque UTE*, vol. 9, no. 1, pp. 127–137, 2018.

[13] G. Chamillard, *UBUNTU: Administración de un sistema Linux*. ENI, 2011.

[14] D. Mancera Bravo, *UF1275 - Selección, instalación, configuración y administracion de los servidores de transferencia de archivos*. 2015.

[15] A. Hernández Jaimes and L. Prada Angarita, “Validation of the statistical characterization of the web server’s network traffic in a university campus as a mechanism of an intrusion detection system,” *Ing. y Desarro.*, vol. 32, no. 1, pp. 64–79, 2014.

[16] M. E. Porter, *La Ventaja Competitiva de las Naciones*. Buenos Aires, 1991.

[17] R. Ingenier and U. C. Issn, “Diseño de un cluster de alta disponibilidad para un entorno educativo virtual universitario,” 2018.

[18] M. Flórez, A. Barbosa Ayala, and E. Muñoz Duarte, “Modelo administrativo para gestión de servidores Linux, implementando mecanismos de seguridad y tecnologías de software libre orientadas a la alta disponibilidad.,” *Rev. UIS Ing.*, vol. 11, no. 2, pp. 227–236, 2012.

[19] D. Teran, *Administración Estratégica de la Función Informática*. 2014.

[20] K. Yaghmour, *Building Embedded Linux Systems*. 2003.

ANEXOS

ANEXO A. INSTALACIÓN Y CONFIGURACIÓN DEL CLÚSTER

La figura 6 nos muestra como editar la tabla de hosts en ambos servidores.

Figura 6 configuración de archivo /etc/hosts.conf

```
192.168.1.20 nodo1 nodo1.linux.com
192.168.1.18 nodo2 nodo2.linux.com
192.168.1.30 vipapache
```

Fuente: Elaboración propia

Los primeros pasos son, la configuración de los dos nodos del clúster con una IP estática, un nombre de host para diferenciar cada nodo y nos aseguramos de que puedan comunicarse entre sí por nombre de nodo. En las figuras 7 y 8 se puede apreciar la prueba de comunicación entre ambos nodos.

Figura 7 ping de nodo 1 hacia nodo 2

```
[root@nodo1 /]# hostname
nodo1
[root@nodo1 /]# ping -c1 nodo2
PING nodo2 (192.168.1.18) 56(84) bytes of data.
64 bytes from nodo2 (192.168.1.18): icmp_seq=1 ttl=64 time=0.525 ms

--- nodo2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.525/0.525/0.525/0.000 ms
```

Fuente: Elaboración propia

Figura 8 ping de nodo 2 hacia nodo 1

```
[root@nodo2 home]# hostname
nodo2
[root@nodo2 home]# ping -c1 nodo1
PING nodo1 (192.168.1.20) 56(84) bytes of data.
64 bytes from nodo1 (192.168.1.20): icmp_seq=1 ttl=64 time=0.248 ms

--- nodo1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.248/0.248/0.248/0.000 ms
```

Fuente: Elaboración propia

Deshabilitamos el firewall en ambos nodos o servidores, con el comando de las figuras 9 y 10.

Figura 9 baja de firewall en nodo 1

```
[root@nodo1 admin]# systemctl stop firewalld
```

Fuente: Elaboración propia

Figura 10 baja de firewall en nodo 2

```
[root@nodo2 home]# systemctl stop firewalld
```

Fuente: Elaboración propia

En el archivo /etc/selinux/config cambiamos a disabled la línea que se refleja en la figura 11. (realizar en ambos nodos)

Figura 11 configuración archivo /etc/selinux/config

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Fuente: Elaboración propia

Instalación de Pacemaker, Corosync y Pcs

Luego haber configurado lo básico, como indican las figuras 12 y 13. procedemos a instalar en ambos nodos los paquetes de los componentes que vamos a usar para configurar el clúster

Figura 12 Descarga de paquetes para clúster en nodo1

```
[root@nodo1 /]# sudo yum install corosync pcs pacemaker
```

Fuente: Elaboración propia

Figura 13 Descarga de paquetes para clúster en nodo2

```
[root@nodo2 home]# sudo yum install corosync pcs pacemaker
```

Fuente: Elaboración propia

Al instalar todos los paquetes, por defecto se creó un usuario *hacluster*, el cual puede ser usado junto con *PCS* para configurar los nodos del clúster, pero primero hay que configurar una contraseña en ambos nodos para ese usuario a través del comando que se observa en las figuras 14 y15.

Figura 14 Configuración de contraseña para hacluster en nodo1

```
[root@nodo1 /]# sudo passwd hacluster
```

Fuente: Elaboración propia

Figura 15 Configuración de contraseña para hacluster en nodo2

```
[root@nodo2 /]# sudo passwd hacluster
```

Fuente: Elaboración propia

Con el comando de las figuras 16 y 17, iniciamos el servicio pcsd en cada nodo.

Figura 16 Inicio de servicio pcsd en nodo1

```
[root@nodo1 /]# sudo systemctl start pcsd
```

Fuente: Elaboración propia

Figura 17 Inicio de servicio pcsd en nodo2

```
[root@nodo2 /]# sudo systemctl start pcsd
```

Fuente: Elaboración propia

Para poder configurar todos los nodos desde un solo punto, debemos autenticarnos en todos los nodos. Lo hacemos de la siguiente manera como indica la figura 18, solo desde un nodo y utilizando el usuario y contraseña de *hacluster*:

Figura 18 Comando para autentificar ambos nodos

```
[root@nodo1 /]# sudo pcs cluster auth nodo1 nodo2
nodo2: Already authorized
nodo1: Already authorized
```

Fuente: Elaboración propia

Creación de Clúster y Agregar Nodos

Iniciamos creando un clúster llamado *cluster_web* y mediante los comandos de la figura 19 agregamos ambos nodos al mismo.

Figura 19 Comando para crear clúster y agregar ambos nodos

```
[root@nodo1 /]# sudo pcs cluster setup --name cluster_web nodo1 nodo2
Destroying cluster on nodes: nodo1, nodo2...
nodo2: Stopping Cluster (pacemaker)...
nodo1: Stopping Cluster (pacemaker)...
nodo2: Successfully destroyed cluster
nodo1: Successfully destroyed cluster

Sending 'pacemaker_remote authkey' to 'nodo1', 'nodo2'
nodo2: successful distribution of the file 'pacemaker_remote authkey'
nodo1: successful distribution of the file 'pacemaker_remote authkey'
Sending cluster config files to the nodes...
nodo1: Succeeded
nodo2: Succeeded

Synchronizing pcsd certificates on nodes nodo1, nodo2...
nodo2: Success
nodo1: Success
Restarting pcsd on the nodes in order to reload the certificates...
nodo2: Success
nodo1: Success
```

Fuente: Elaboración propia

Una vez creado el clúster y agregado los nodos, se lo puede iniciar como vemos en la figura 20, pero no hará mucho debido a que no tiene ningún recurso configurado.

Figura 20 comando para iniciar clúster

```
[root@nodo1 /]# pcs cluster start --all
nodo1: Starting Cluster (corosync)...
nodo2: Starting Cluster (corosync)...
nodo1: Starting Cluster (pacemaker)...
nodo2: Starting Cluster (pacemaker)...
```

Fuente: Elaboración propia

En la figura 21 verificamos el estado del clúster

Figura 21 comando para verificar estado de clúster

```
[root@nodo1 /]# pcs status cluster
Cluster Status:
 Stack: corosync
 Current DC: nodo1 (version 1.1.19-8.el7_6.4-c3c624ea3d) - partition with quorum
 Last updated: Wed Jul 24 16:43:15 2019
 Last change: Wed Jul 24 15:42:33 2019 by root via cibadmin on nodo1
 2 nodes configured
 2 resources configured

PCSD Status:
  nodo2: Online
  nodo1: Online
```

Fuente: Elaboración propia

Las figuras 22 y 23 nos muestran otra manera de verificar el estado de los nodos del clúster.

Figura 22 comando para verificar estado de los nodos

```
[root@nodo1 /]# pcs status nodes
Pacemaker Nodes:
 Online: nodo1 nodo2
 Standby:
 Maintenance:
 Offline:
Pacemaker Remote Nodes:
 Online:
 Standby:
 Maintenance:
 Offline:
```

Fuente: Elaboración propia

Figura 23 comando para verificar estado de los nodos (corosync)

```
[root@nodo1 /]# pcs status corosync

Membership information
----------------------
    Nodeid      Votes Name
         2          1 nodo2
         1          1 nodo1 (local)
```

Fuente: Elaboración propia

El STONITH es un mecanismo que nos garantiza no terminar con dos nodos que crean que están activos. En este caso como tenemos un clúster simple, deshabilitaremos esta opción con el comando de la figura 24.

Figura 24 comando para deshabilitar STONITH

```
[root@nodo1 /]# pcs property set stonith-enable=false
```

Fuente: Elaboración propia

El quórum describe el número mínimo de nodos en el clúster que deben estar activos para que el clúster esté disponible. Por defecto, el quórum se considera demasiado bajo si el número total de nodos es menor que el doble del número de nodos activos. Para nuestro clúster de dos nodos, significa que ambos nodos deben estar disponibles para que el clúster esté disponible, por lo tanto, procedemos a ignorar el quorum de la forma que nos indica la figura 25.

Figura 25 comando para ignorar el quorum

```
[root@nodo1 /]# pcs property set no-quorum-policy=ignore
```

Fuente: Elaboración propia

ANEXO B. CONFIGURACIÓN DE DIRECCIÓN IP VIRTUAL (VIP)

Para que nuestro clúster haga algo, agregamos al mismo una IP virtual como recurso. Esta dirección IP es la que nos permitirá acceder al servidor web. La figura 26 nos indica el comando necesario para agregar el recurso.

Figura 26 comando para agregar recurso de dirección VIP

```
[root@nodo1 /]# pcs create vip_apache ocf:heartbeat:IPaddr2 ip=192.168.1.30 cidr_netmask=24
op monitor interval=30s
```

Fuente: Elaboración propia

Comprobamos si existe comunicación con la dirección IP virtual, haciendo un ping como está en la figura 27.

Figura 27 ping de comunicación con IP virtual

```
[root@nodo1 /]# ping vip_apache
PING vip_apache (192.168.1.30) 56(84) bytes of data.
64 bytes from vip_apache (192.168.1.30): icmp_seq=1 ttl=64 time=0.015 ms
64 bytes from vip_apache (192.168.1.30): icmp_seq=2 ttl=64 time=0.027 ms
64 bytes from vip_apache (192.168.1.30): icmp_seq=3 ttl=64 time=0.023 ms
64 bytes from vip_apache (192.168.1.30): icmp_seq=4 ttl=64 time=0.025 ms
^C
--- vip_apache ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.015/0.022/0.027/0.006 ms
```

Fuente: Elaboración propia

ANEXO C. INSTALACIÓN Y CONFIGURACIÓN DE SERVICIO WEB APACHE

Comenzamos descargando el paquete *httpd* para poder configurar el servicio de Apache. Las figuras 28 y 29 nos indican el comando necesario.

Figura 28 instalación de Apache en nodo1

```
[root@nodo1 /]# sudo yum install httpd
```

Fuente: Elaboración propia

Figura 29 instalación de Apache en nodo2

```
[root@nodo2 /]# sudo yum install httpd
```

Fuente: Elaboración propia

Para que el clúster verifique si Apache todavía está activo y responde en el nodo activo, necesitamos crear un pequeño mecanismo de prueba. Para esto, agregaremos una página de estado que será consultada regularmente.

Creamos el archivo */etc/httpd/conf.d/serverstatus.conf* y se agrega el siguiente contenido de la figura 30. (realizar este paso en ambos servidores)

Figura 30 configuración de archivo /etc/httpd/conf.d/serverstatus.conf

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
Listen 127.0.0.1:80
<Location /server-status>
        SetHandler server-status
        Order deny,allow
        Deny from all
        Allow from 127.0.0.1
</Location>
```

Fuente: Elaboración propia

Desactivamos el estado actual del Listen en la configuración de Apache para evitar intentar escuchar varias veces en el mismo puerto según las figuras 31 y 32.

Figura 31 comando para desactviar Listen de Apache en nodo1

```
[root@nodo1 /]# sudo sed -i 's/Listen/#Listen/' /etc/httpd/conf/httpd.conf
```

Fuente: Elaboración propia

Figura 32 comando para desactviar Listen de Apache en nodo2

```
[root@nodo2 /]# sudo sed -i 's/Listen/#Listen/' /etc/httpd/conf/httpd.conf
```

Fuente: Elaboración propia

Reiniciamos el servicio de Apache y verificamos si la página funciona con los comandos de la figura 33. (realizar este paso en ambos nodos).

Figura 33 reinicio de Apache y verificación de la página de estado

```
[root@nodo1 /]# sudo systemctl restart httpd
[root@nodo1 /]# wget http://127.0.0.1/server-status
--2019-07-13 16:37:15--  http://127.0.0.1/server-status
Conectando con 127.0.0.1:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 2738 (2,7K) [text/html]
Grabando a: “server-status”

100%[======================================================================>] 2.738       --.-K/s   en 0s

2019-07-13 16:37:15 (363 MB/s) - “server-status” guardado [2738/2738]
```

Fuente: Elaboración propia

Ingresamos la página web dentro del directorio */var/www/html/* con el contenido de las figuras 34 y 35.

Figura 34 contenido de página web en nodo1

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
<html>
 <h1>Bienvenido al nodo1 principal</h1>
</html>
```

Fuente: Elaboración propia

Figura 35 contenido de página web en nodo2

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
<html>
 <h1>Bienvenido al nodo espejo</h1>
</html>
```

Fuente: Elaboración propia

Para permitir que el clúster controle Apache, primero hay que detener el servicio web en cada nodo. Luego se configura el Listen de Apache para que escuche la dirección IP virtual. La figura 36 nos muestra los comandos necesarios para este proceso.

Figura 36 configuración de Apache con dirección IP virtual en nodo1

```
[root@nodo1 ~]# systemctl stop httpd
[root@nodo1 ~]# echo "Listen 192.168.1.30:80"|sudo tee --append /etc/httpd/conf/httpd.conf
Listen 192.168.1.30:80
```

Fuente: Elaboración propia

Figura 37 configuración de Apache con dirección IP virtual en nodo2

```
[root@nodo2 ~]# systemctl stop httpd
[root@nodo2 ~]# echo "Listen 192.168.1.30:80"|sudo tee --append /etc/httpd/conf/httpd.conf
Listen 192.168.1.30:80
```

Fuente: Elaboración propia

Una vez que Apache puede ser controlado por el clúster, ya se puede agregar el recurso para el servidor web, solo es necesario de hacerlo en un nodo a través de PCS como en la figura 38.

Figura 38 comando para agregar un recurso webserver al clúster

```
[root@nodo1 /]# pcs resource create webserver ocf:heartbeat:apache configfile=/etc/httpd/conf/httpd.conf
statusurl="http://localhost/server-status" op monitor interval=60s
```

Fuente: Elaboración propia

Configuramos el orden y restricciones de los servicios como en la figura 39, para que la IP virtual no inicie en un nodo diferente del servidor web y así evitar fallas.

Figura 39 configuración de restricciones para IP virutal y webserver

```
[root@nodo1 ~]# sudo pcs constraint colocation add webserver vipapache INFINITY
[root@nodo1 ~]# sudo pcs constraint order vipapache then webserver
Adding vipapache webserver (kind: Mandatory) (Options: first-action=start then-action=start)
```

Fuente: Elaboración propia

En la figura 40 asignamos la preferencia al grupo de servicios sobre un nodo (por defecto es infinito).

Figura 40 configuración de preferencia de nodo de ejecución

```
[root@nodo1 ~]# sudo pcs constraint location webserver prefers nodo1=50
```

Fuente: Elaboración propia

Finalmente verificamos todas las restricciones se lo puede hacer mediante el comando de la figura 41.

Figura 41 comando para verificar restricciones del clúster

```
[root@nodo1 /]# pcs constraint
Location Constraints:
  Resource: webserver
    Enabled on: nodo1 (score:50)
Ordering Constraints:
  start vip_apache then start webserver (kind:Mandatory)
Colocation Constraints:
  webserver with vip_apache (score:INFINITY)
Ticket Constraints:
```

Fuente: Elaboración propia

Las figuras 42 y 43 nos indican como reiniciar el clúster y verificar toda la configuración.

Figura 42 comando para reiniciar clúster

```
[root@nodo1 /]# pcs cluster stop --all && pcs cluster start --all
nodo2: Stopping Cluster (pacemaker)...
nodo1: Stopping Cluster (pacemaker)...
nodo2: Stopping Cluster (corosync)...
nodo1: Stopping Cluster (corosync)...
nodo1: Starting Cluster (corosync)...
nodo2: Starting Cluster (corosync)...
nodo1: Starting Cluster (pacemaker)...
nodo2: Starting Cluster (pacemaker)...
```

Fuente: Elaboración propia

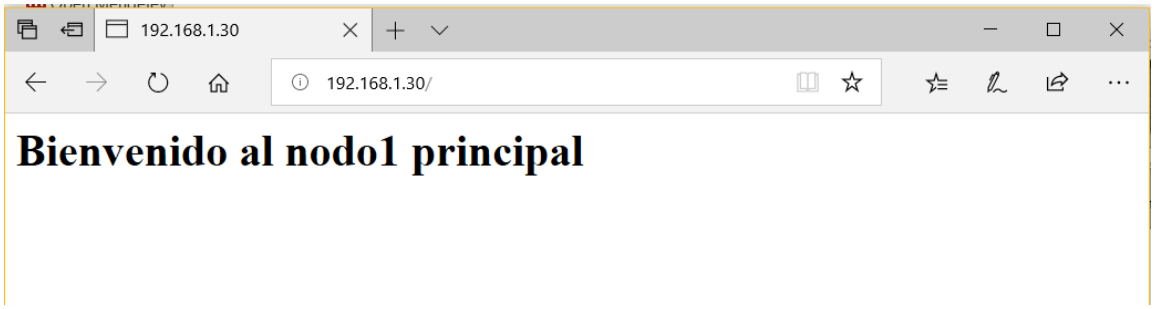
Figura 43 verificación de estado final del clúster

```
[root@nodo1 /]# pcs status
Cluster name: cluster_web
Stack: corosync
Current DC: nodo1 (version 1.1.19-8.el7_6.4-c3c624ea3d) - partition with quorum
Last updated: Wed Jul 24 17:52:38 2019
Last change: Wed Jul 24 16:42:25 2019 by root via cibadmin on nodo1

2 nodes configured
2 resources configured

Online: [ nodo1 nodo2 ]

Full list of resources:

 vip_apache     (ocf::heartbeat:IPaddr2):       Started nodo1
 webserver      (ocf::heartbeat:apache):        Started nodo1

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

Fuente: Elaboración propia

Como podemos ver, el estado actual del clúster ejecuta el servidor web y la IP virtual en el nodo1. En la figura 44 se puede acceder con la dirección IP virtual 192.168.1.30.

Figura 44 vista de página web ejecutada en nodo1 (principal)



Fuente: Elaboración propia

Ahora para hacer una prueba de la replicación del servicio web, detenemos el nodo1 del clúster y verificamos si la página web se mantiene disponible. La figura 45 nos indica el comando para detener el nodo 1 y en la figura 46 se puede verificar el estado *offline* del mismo nodo.

Figura 45 comando para detener nodo 1

```
[root@nodo1 /]# pcs cluster stop nodo1
nodo1: Stopping Cluster (pacemaker)...
nodo1: Stopping Cluster (corosync)...
```

Fuente: Elaboración propia

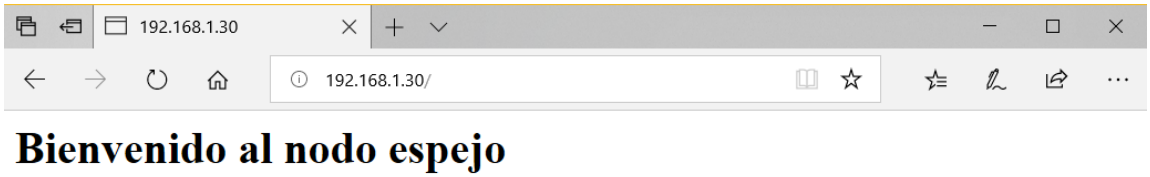
Figura 46 verificación de estado de clúster con nodo 1 detenido

```
[root@nodo2 /]# pcs status
Cluster name: cluster_web
Stack: corosync
Current DC: nodo2 (version 1.1.19-8.el7_6.4-c3c624ea3d) - partition with quorum
Last updated: Wed Jul 24 18:36:21 2019
Last change: Wed Jul 24 16:42:25 2019 by root via cibadmin on nodo1

2 nodes configured
2 resources configured

Online: [ nodo2 ]
OFFLINE: [ nodo1 ]

Full list of resources:

 vip_apache     (ocf::heartbeat:IPaddr2):       Started nodo2
 webserver      (ocf::heartbeat:apache):        Started nodo2

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

Fuente: Elaboración propia

Actualizamos la misma URL con la misma dirección IP y podemos ver que el resultado es la página web servida y ejecutada por el nodo2.

Figura 47 vista de página web ejecutada en nodo2



Fuente: Elaboración propia

ANEXO D. INSTALACIÓN Y CONFIGURACIÓN DE RSYNC

Instalamos los paquetes necesarios para poder utilizar rsync mediante el comando de la figura 48.

Figura 48 instalación de paquetes de rsync

```
[root@nodo1 /]# yum install xinetd rsync
```

Fuente: Elaboración propia

Activamos el servicio rsync dentro de xinetd, editando el archivo */etc/xinetd.d/rsync* como se muestra en la figura 49.

Figura 49 configuración de archivo /etc/xinetd.d/rsync

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
# default: off
# description: The rsync server is a good addition to an ftp server, as it \
#       allows crc checksumming etc.
services rsync
{
        disable = no
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/bin/rsync
        server_args     = --daemon
        log_on_failure  += USERID
}
```

Fuente: Elaboración propia

Configuramos el servicio de rsync editando el archivo de la figura 50:

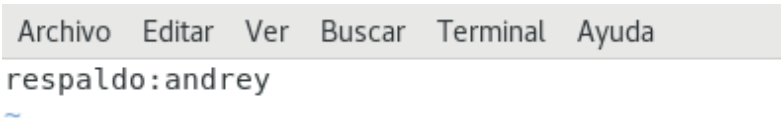
Figura 50 configuración de archivo /etc/rsyncd.conf

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
# /etc/rsyncd: configuration file for rsync daemon mode

# See rsyncd.conf man page for more options.

# configuration example:

# uid = nobody
# gid = nobody
# use chroot = yes
# max connections = 4
 log file = /var/log/rsyncd.log
 pid file = /var/run/rsyncd.pid
 lock file = /var/run/rsync.lock

 uid = nobody
 gid = nobody
 read only = no
 hosts allow = 192.168.1.1/255.255.255.0
 auth users = respaldo
 secrets file = /etc/rsync.secrets

# exclude = lost+found/
# transfer logging = yes
# timeout = 900
# ignore nonreadable = yes
# dont compress   = *.gz *.tgz *.zip *.z *.Z *.rpm *.deb *.bz2

 [documentos]
        path = /home/admin
        comment = home del usuario admin
```

Fuente: Elaboración propia

Agregamos el usuario y contraseña del usuario del servidor remoto al cual vamos a acceder, seguido de dos puntos como en la figura 51.

Figura 51 contenido de archivo /etc/rsyncd.secrets



Fuente: Elaboración propia

Para todos los archivos rsync se cambia el propietario y grupo a root y también los permisos para el propietario, según nos indica la figura 52.

*Figura 52 modificación de propietario y permisos de archivos /etc/rsync**

```
[root@nodo1 ~]# chown root.root /etc/rsync*
[root@nodo1 ~]# chmod 600 /etc/rsync*
```

Fuente: Elaboración propia

Reiniciamos el xinetd con el comando especificado en la figura 53.

Figura 53 reinicio de xinetd

```
[root@nodo1 /]# service xinetd restart
Redirecting to /bin/systemctl restart xinetd.service
```

Fuente: Elaboración propia

Habilitamos el puerto ssh del firewalld de manera permanente con el comando de la figura 54.

Figura 54 permiso permanente para ssh en firewall

```
[root@localhost compartir]# systemctl start firewalld
[root@localhost compartir]# sudo firewall-cmd --permanent --add-service=ssh
Warning: ALREADY_ENABLED: ssh
success
```

Fuente: Elaboración propia

Instalación de llave pública y privada para acceso seguro mediante ssh

Para poder configurar el envío de archivos remotos automatizados, se lo puede hacer mediante ssh con rsync, pero es necesario configurar una llave pública y privada para evitar que nos pida contraseña todas las veces que se ejecute el comando.

Con el comando de la figura 55 podemos instalar la llave pública y privada para hacer uso de ssh sin necesidad de usar contraseña, presionamos *enter* en todas las preguntas. Nuestra llave publica estará en *home/usuario/.ssh/id_dsa.pub*.

Figura 55 comando para instalar llave pública y privada

```
[root@nodo1 /]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:kzFsdYTn7UKq6wktVTfWzTIF6LRbBbU1mnylYVgD3cU root@vip_apache
The key's randomart image is:
+---[RSA 2048]----+
|           .o+*BBB|
|       . ...*o+BE|
|        = .=+*=+o|
|       . = o=.+o |
|        S  o +   |
|       o .. o .  |
|      o ..   .   |
|       o..       |
|       .+.       |
+----[SHA256]-----+
```

Fuente: Elaboración propia

Utilizando el comando de la figura 56, copiamos esta llave generada al servidor espejo con dirección IP: 192.168.1.18

Figura 56 copia de llave pública en servidor espejo

```
[root@nodo1 /]# ssh-copy-id respaldo@192.168.1.18
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that
 are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is
 to install the new keys
respaldo@192.168.1.18's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'respaldo@192.168.1.18'"
and check to make sure that only the key(s) you wanted were added.
```

Fuente: Elaboración propia

Para que funcione correctamente la clave pública, al directorio oculto /.ssh en el servidor espejo, otorgamos todos los permisos sólo para el usuario. Realizamos a través del comando chmod de la figura 57.

Figura 57 cambio de permisos para directorio /.ssh

```
[respaldo@localhost ~]$ chmod 700 .ssh
```

Fuente: Elaboración propia

En la figura 58 probamos el envío seguro de archivos con rsync mediante ssh, sólo la primera vez nos pedirá contraseña de acceso.

Figura 58 envío de archivos con rsync mediante ssh

```
[root@nodo1 ~]# rsync -avz -e ssh /home/admin/Imágenes/ respaldo@192.168.1.18:/h
ome/respaldo/Imágenes/
sending incremental file list

sent 192 bytes  received 12 bytes  136.00 bytes/sec
total size is 781,423  speedup is 3,830.50
```

Fuente: Elaboración propia

La figura 59 nos indica el comando para ingresar al crontab del usuario en el cual posteriormente se agregarán tareas, en este caso los comandos rsync, para que se ejecuten frecuentemente.

Figura 59 modificación de comando crontab

```
[admin@localhost /]$ crontab -e
crontab: installing new crontab
```

Fuente: Elaboración propia

ANEXO E. SINCRONIZACIÓN DE DIRECTORIOS

Rsnapshot es una colección de scripts que automatizan el proceso de crear copias de seguridad incrementales, su principal ventaja es que permite crear diferentes "snapshots" (capturas de la data).

Para no tener problemas de permisos, creamos como usuario root una carpeta de nombre *sinc* en la ruta que nos indica la figura 60.

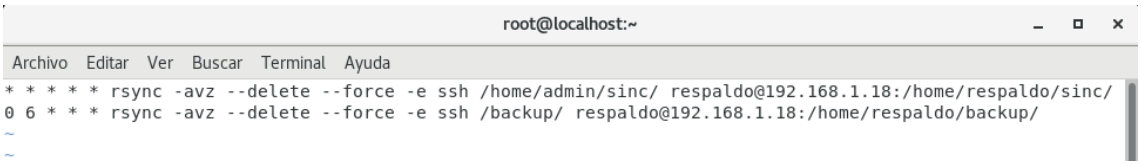
Figura 60 creación de directorio /home/admin/sinc de sincronización

```
[root@localhost admin]# mkdir sinc
```

Fuente: Elaboración propia

Agregamos al cron del usuario los siguientes comandos rsync para sincronizar las carpetas con la información más importante para la empresa. En este caso sería los directorios */home/admin/sinc* y */backup/* con la periodicidad que indica la figura 61.

Figura 61 configuración de crontab -e del usuario



Fuente: Elaboración propia

ANEXO F. INSTALACIÓN Y CONFIGURACIÓN DE RSNAPSHOT

Como nos indica la figura 62, primero se debe instalar el repositorio epel, el mismo que contiene los paquetes para luego poder instalar rsnapshot.

Figura 62 Instalación de repositorio epel-release

```
[root@localhost ~]# yum install epel-release
Complementos cargados:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.upb.edu.co
 * extras: mirror.ci.ifes.edu.br
 * updates: mirror.globo.com
Resolviendo dependencias
--> Ejecutando prueba de transacción
---> Paquete epel-release.noarch 0:7-11 debe ser instalado
--> Resolución de dependencias finalizada
```

Fuente: Elaboración propia

Ahora si procedemos a descargar la herramienta rsnapshot con el comando de la figura 63.

Figura 63 Instalación de paquete rsnapshot

```
[root@localhost ~]# yum install rsnapshot
Complementos cargados:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.upb.edu.co
 * epel: mirror.globo.com
 * extras: mirror.ci.ifes.edu.br
 * updates: mirror.globo.com
Resolviendo dependencias
--> Ejecutando prueba de transacción
---> Paquete rsnapshot.noarch 0:1.4.2-2.el7 debe ser instalado
--> Resolución de dependencias finalizada

Dependencias resueltas
```

Fuente: Elaboración propia

Configuramos el archivo de rsnapshot modificando solo los siguientes parámetros:

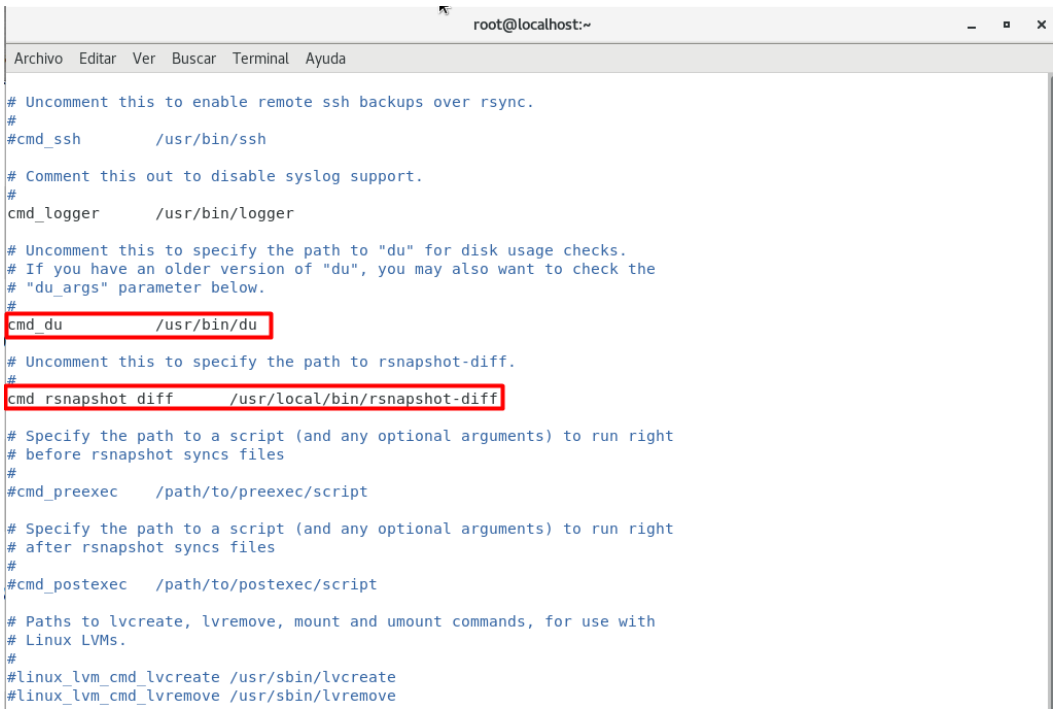
- En la figura 64 el parámetro *snapshot_root,* indica la ruta del directorio donde se almacenarán los respaldos. En este caso se ha creado una carpeta de nombre */backup*.

Figura 64 configuración de parámetro snapshot_root en archivo /etc/rsnapshot.conf

Fuente: Elaboración propia

- Los parámetros *cmd_du* y *cmd_rsnapshot_diff* de la figura 65, se utilizan para saber el espacio ocupado de los respaldos y ver las diferencias entre los respaldos de seguridad respectivamente.

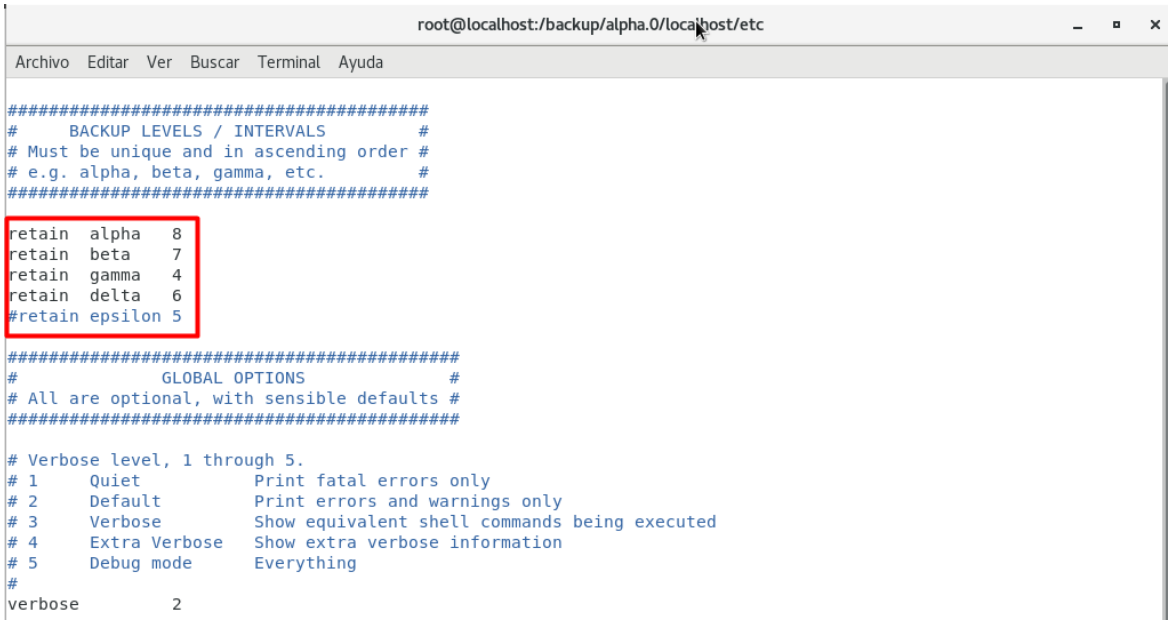
Figura 65 configuración de parámetros cmd_du y cmd_rsnpashot_diff



Fuente: Elaboración propia

- Para establecer el intervalo de los backup, editamos el parámetro *retain* como se aprecia en la figura 66: 8 últimas horas (alpha), 7 últimos días (beta), 4 últimas semanas (gamma) y 6 últimos meses (delta).

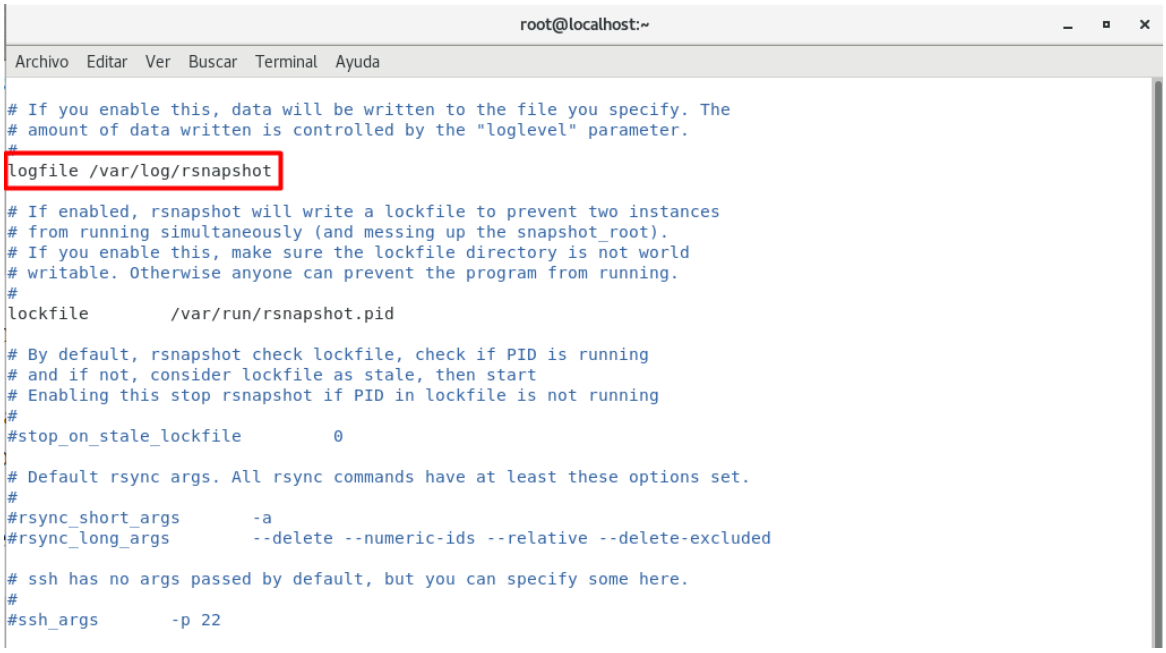
Figura 66 configuración de parámetros retain en archivo /etc/rsnapshot.conf



Fuente: Elaboración propia

- En la figura 67, con el parámetro *logfile* asignamos el directorio donde estará el archivo log de rsnapshot.

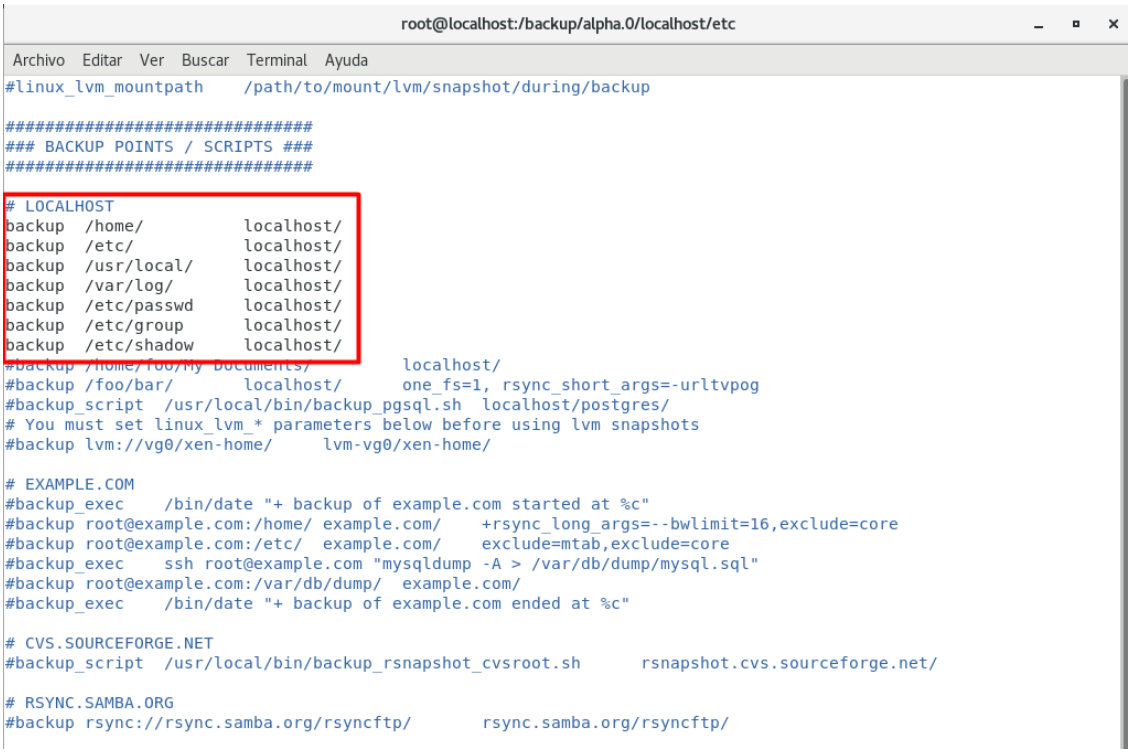
Figura 67 configuración de parámetro logfile en archivo /etc/rsnapshot.conf



Fuente: Elaboración propia

- El parámetro *backup de la figura 68,* nos sirve para indicar cuales carpetas se desea salvar con una copia de seguridad. Después de *backup* se indica el directorio a salvar y luego *localhost/* que será el nombre de la carpeta donde se almacena el backup.

Figura 68 configuración de parámetros backup del archivo /etc/rsnapshot.conf



Fuente: Elaboración propia

Con el comando de la figura 69 comprobamos que la configuración de rsnapshot esté correcta.

Figura 69 test de configuración de archivo /etc/rsnapshot.conf

```
[root@localhost ~]# sudo rsnapshot configtest
Syntax OK
```

Fuente: Elaboración propia

Comprobamos que funciona realizando nuestra primera copia de seguridad mediante el comando de la figura 70.

Figura 70 prueba de respaldo por hora con rsnapshot

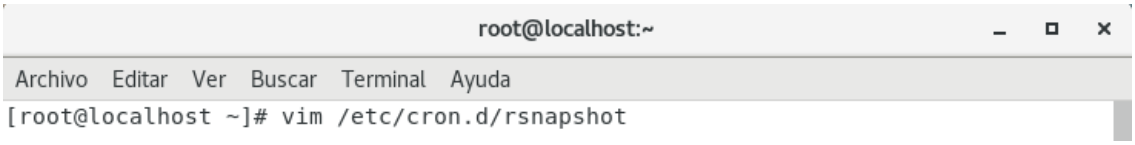
```
[root@localhost ~]# rsnapshot alpha
[root@localhost ~]# ls /backup
alpha.0  alpha.1
```

Fuente: Elaboración propia

Copias de Seguridad Automatizadas

Las copias automáticas se ejecutan mediante el propio archivo de configuración de rsnapshot para cron, o también a través del cron de Linux. La figura 71 nos indica el acceso para su configuración.

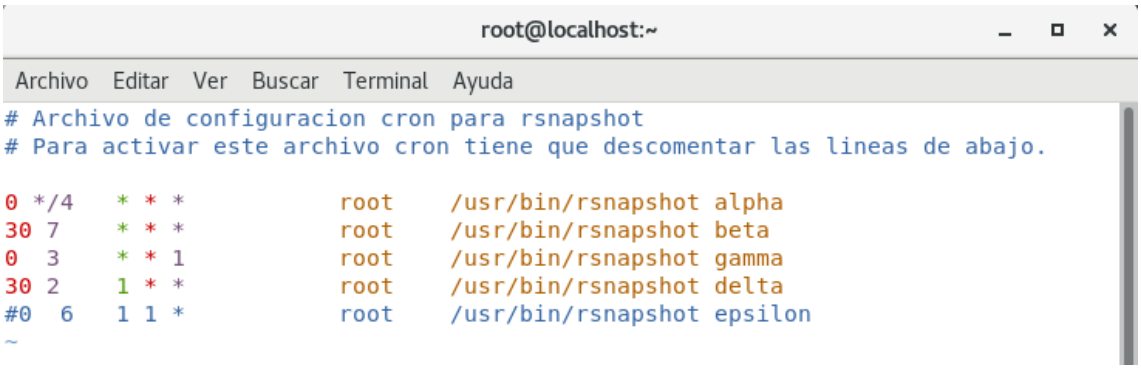
Figura 71 comando para editar archivo cron /etc/cron.d/rsnapshot



Fuente: Elaboración propia

Editamos el archivo como nos muestra la figura 72.

Figura 72 edición de archivo cron de rsnapshot



Fuente: Elaboración propia