DISEÑO E IMPLEMENTACIÓN DE UN SERVICIO DE CORREO EMPRESARIAL ROBUSTO USANDO LA HERRAMIENTA ZIMBRA COLLABORATION OPEN SOURCE EDITION VERSIÓN 8.6.

DIEGO ARMANDO ARIZA ORTIZ
SELIM ALBERTO SABA SANTIAGO

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERIA ELÉCTRICA, ELECTRÓNICA
Y DE TELECOMUNICACIONES
ESPECIALIZACIÓN EN TELECOMUNICACIONES
BUCARAMANGA
2016

DISEÑO E IMPLEMENTACIÓN DE UN SERVICIO DE CORREO EMPRESARIAL ROBUSTO USANDO LA HERRAMIENTA ZIMBRA COLLABORATION OPEN SOURCE EDITION VERSIÓN 8.6.

DIEGO ARMANDO ARIZA ORTIZ
SELIM ALBERTO SABA SANTIAGO

Trabajo de aplicación presentado como requisito para optar al título de:
ESPECIALISTA EN TELECOMUNICACIONES

Director:
Mg. Pedro Javier Trujillo Tarazona

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERIA ELÉCTRICA, ELECTRÓNICA
Y DE TELECOMUNICACIONES
ESPECIALIZACIÓN EN TELECOMUNICACIONES
BUCARAMANGA
2016

CONTENIDO

pág.

TABLA DE ILUSTRACIONES

pág.

LISTA DE ABREVIATURAS

DNS	Domain Name System (sistema de nombre de dominio)
FQDN	Fully Qualified Domain Name (nombre de dominio totalmente cualificado)
IMAP	Internet Messages Access Protocol (protocolo de accesso de mensajes de internet)
LMTP	Local Mail Transfer Protocol (protocolo de trasferencia de correo local)
MTA	Mail Transfer Agent (agente de transferencia de correo)
PDU	Power Distribution Power (unidad de distribución de energia)
POP	Post Office Protocol (protocolo de oficina de correo)
RBL	Realtime Blackhole List
SMTP	Simple Mail Transfer Protocol (protocolo de transferencia de correo simple)
SOAP	Simple Object Access Protocol (protocolo de acceso a objeto simple)
SPF	Sender Policy Framework (convenio de remitentes)
SSL	Secure Socket Layer (capa de puertos seguros)
TLS	Transport Layer Security (seguridad de la capa de transporte)
UPS	Uninterruptible Power Supply (fuente ininterrumpida de energía)
XML	eXtensible Markup Language (lenguaje de marcas extensible)

RESUMEN

TÍTULO: DISEÑO E IMPLEMENTACIÓN DE UN SERVICIO DE CORREO EMPRESARIAL ROBUSTO USANDO LA HERRAMIENTA ZIMBRA COLLABORATION OPEN SOURCE EDITION VERSIÓN 8.6.[*]

AUTORES: Diego Armando Ariza Ortiz, Selim Alberto Saba Santiago[†]

PALABRAS CLAVE: Alta disponibilidad, correo electrónico, DRBD, Pacemaker, replicación de datos, seguridad, Zimbra.

DESCRIPCIÓN:

El correo electrónico cumple un papel primordial dentro de las herramientas empresariales actuales, debido a su capacidad de manejar información de forma fácil y eficiente. Para asegurar el continuo funcionamiento de este servicio en Offimedicas S.A. se aplicaron prácticas de respaldo de datos utilizando equipos servidores con los que ya contaba la empresa. Para emplear dichas políticas se utilizaron herramientas de software libre como DRBD y Pacemaker, las cuales se encargan de duplicar la información requerida en los equipos dispuestos para esta tarea en tiempo real, además de elegir, entre el grupo de servidores disponibles, cual es el más apto para prestar el servicio. Por otra parte, se eligió Zimbra Collaboration Open Source Edition versión 8.6 como software de manejo para el servicio de correo. Dentro de sus parámetros de configuración, se hizo énfasis en las secciones de seguridad en donde se aplicaron reglas para disminuir riesgos de infección por virus en los equipos clientes y se seleccionaron medios de comunicación con datos cifrados para proteger la integridad de la información. La implementación de este proyecto se puso a prueba en el ambiente real obteniendo los resultados satisfactorios para la alta disponibilidad y reducción de recepción de mensajes con posibles amenazas para los equipos.

[*] Trabajo de aplicación

[†] Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Especialización en Telecomunicaciones. Director: M. Sc. Pedro Javier Trujillo Tarazona

SUMMARY

TITLE: DESIGN AND IMPLEMENT A ROBUST CORPORATE MAILING SYSTEM USING ZIMBRA COLLABORATION OPEN SOURCE EDITION V. 8.6 AS A TOOL.[*]

AUTHORS: Diego Armando Ariza Ortiz, Selim Alberto Saba Santiago.[†]

KEYWORDS: data replication, DRBD, email, High availability, Pacemaker, security, Zimbra.

DESCRIPTION:

The email servers accomplishes a primordial role in contemporary corporate tools on account of their ability to manage information in an easy and efficient way. In order to ensure the uninterrupted functioning of this system in Offimedicas S.A. data-backup techniques were applied using servers which the company already owned. In order to employ said politics there were used open software tools like DRBD and Pacemaker, which duplicate the required information in the provided equipment for this task, in real time, in addition to choosing, amongst the group of available servers, the most fit to offer service, which is the server with more free resources like RAM memory and idle processor. On the other hand, Zimbra Collaboration Open Source Edition version 8.6 was chosen as the email management software. In its configuration parameters it is emphasized in security sections where rules were applied to diminish the risk of infection by viruses on the client computers and the selection of communication means with encrypted data to protect the integrity of the information. The implementation of this project was tested in a real environment with satisfactory results for high availability as well as reception reduction for messages which carry possible threats to the equipment.

[*] Aplied work

[†] Faculty of Physical-Mechanical Engineerings. School of Electrical Engineering, Electronic and Telecommunication. Specialization in Telecommunication. Director: M. Sc. Pedro Javier Trujillo Tarazona.

INTRODUCCIÓN

Para cualquier empresa el correo electrónico es una herramienta indispensable para el desarrollo de sus labores diarias, especialmente en Offimedicas S.A. que lo requiere funcionando ininterrumpidamente para los procesos de venta, facturación, compras entre otros. Debido a su funcionalidad, este servicio debe estar disponible la mayor cantidad de tiempo posible, porque la ausencia de él afecta directamente la productividad de la compañía. Actualmente Offimedicas no cuenta con ningún plan de respaldo que disminuya el impacto de una posible falla en este servicio. Para evitar la interrupción en la prestación se debe reducir al mínimo las causas de fallos y adicionalmente se necesita tener planes de contingencia en caso de una emergencia.

Este proyecto de aplicación tiene como objetivo implementar algunas herramientas de código abierto para establecer un servicio de correo robusto a un bajo costo, aprovechando las ventajas de manejarlo internamente, tal como la autonomía de administración del sistema. Para lograr el objetivo se usan los equipos disponibles en la empresa, se aplican estrategias de prevención y se diseñan planes de emergencia.

Los planes de emergencia se fundamentan en la redundancia de información, de tal forma que, en caso de falla o ausencia de uno de los equipos, los datos permanezcan disponibles y el servidor en funcionamiento pueda tomar el papel principal y las estrategias de prevención son reglas de seguridad que disminuye el flujo de mensajes no deseados y de mensajes con software maliciosos, además de cifrar los datos en el envío de correos.

Para la implementación de las herramientas DRBD y Pacemaker, encargadas de duplicación y sincronización de datos entre los equipos del cluster de alta disponibilidad, se basó la investigación en la documentación que se encuentra en las páginas web oficiales de estas aplicaciones. Para la instalación e implementación de reglas de seguridad en Zimbra, además de la documentación disponible en la página oficial, fue necesario hacer uso de la ayuda que se encuentra en la interfaz gráfica de la consola de administración de este software.

Con el proyecto se logra el objetivo principal de diseñar un servicio de correo robusto, pero está pendiente la transición del correo actual al nuevo. Esto se debe a la necesidad de encontrar una forma eficaz de copiar toda la información existente del servidor actual al cluster de alta disponibilidad. Esta tarea está por fuera del alcance de este proyecto y será desarrollada a lo largo de los meses de julio y agosto del presente año.

En el primer capítulo del informe, se recopila el marco conceptual referente a las tecnologías aplicadas. En el segundo apartado se hace una breve descripción del entorno donde se aplica el proyecto y los elementos utilizados. En los capítulos 3, 4, 5 y 6, se muestran los procedimientos necesarios para desarrollar el cluster de alta disponibilidad para el servicio de correo. En la última parte se hace una descripción de los parámetros que se pueden ajustar para fortalecer la seguridad del correo electrónico basado en Zimbra.

1 MARCO CONCEPTUAL

1.1 SMTP (PROTOCOLO SIMPLE DE TRANSFERENCIA DE CORREO)

Es un estándar de Internet para el intercambio de correo electrónico entre dispositivos como computadores o teléfonos móviles.

Una característica importante de SMTP es la capacidad de transportar correo a través de múltiples redes, fenómeno normalmente conocido como "*SMTP mail relaying"* (Klensin, 2008). De esta forma, un mensaje de correo puede pasar por un número de relevos intermedios en su camino desde el origen hasta el destinatario. Los registros MX de un sistema de nombres de dominio, son usados para identificar los siguientes saltos apara un mensaje que es transportado.

Cuando un cliente SMTP tiene un mensaje que transmitir, establece un canal de dos vías con el servidor SMTP. El servidor SMTP puede ser el destino final, o un relevo intermediario (lo que significa que asume el papel de cliente SMTP luego de recibir el mensaje) o de "*GATEWAY"* (que puede transportar el mensaje usando un protocolo diferente a SMTP). Los comandos son generados por el cliente y enviados al servidor y las respuestas son devueltas por el servidor de acuerdo a los comandos.

1.2 MIME (EXTENSIONES MULTIPROPOSITO DE CORREO DE INTERNET)

El formato de mensaje estándar especificado por SMTP no permite que las líneas tengan más de 1000 caracteres ASCII. Esto quiere decir que si se basa solo en SMTP no se podría incluir imágenes o texto con formato en correo electrónico (Freed & Borenstein, 1996). Esto era suficiente en los primeros momentos del Internet, sin embargo, para superar estas limitaciones, la IETF comenzó a desarrollar MIME en 1991.

MIME es un estándar para codificar e interpretar archivos binarios, imágenes, vídeos y conjuntos de caracteres que no pertenecen al estándar ASCII, en un mensaje de correo, identificando cada elemento de un mensaje de correo de acuerdo al tipo de contenido. Un tipo de contenido es "*multipart",* cuando tiene más de un elemento que no pertenece al estándar ASCII, por ejemplo, un mensaje con algo de contenido de texto, un archivo binario y una imagen.

1.3 POP (PROTOCOLO DE OFICINA DE CORREO O "PROTOCOLO DE OFICINA POSTAL")

Es un protocolo de la capa de aplicación usado para obtener mensajes de un servidor de correo. La versión más actual y usada es POP3, la cual trabaja sobre TCP en el puerto 110 (Myers, Mellon, & Rose, 1996). Con POP3 los mensajes son almacenados en servidor de correo hasta que un usuario se conecta por medio de un cliente de correo para obtener los correos, Cuando el cliente obtiene los correos, los mensajes son descargados a la máquina y luego eliminados del servidor.

Unas de las ventajas de POP3 es que minimiza el uso de recursos porque elimina el correo luego descargarlo y si tiene una conexión lenta, los mensajes de correos se descargan a lo largo del tiempo y cuando el usuario quiere ver los mensajes no debe esperar a que se descarguen. Adicionalmente, virtualmente todos los servidores y clientes de correo lo soportan. Sin embargo, el hecho de borrar los correos al momento de leerlos puede ser una desventaja en algunos casos. Este protocolo cabe

mejor para usuarios que guardan su correo en una sola máquina todo el tiempo. Los usuarios que cambian de máquina tienen la desventaja de no poder tener el correo sincronizado en cada máquina.

Un servidor POP escucha en el puerto TCP 110 y la comunicación encriptada es solicitada luego de haber inicializado el protocolo, usando el comando STLS si lo soporta. Por otra parte, la comunicación también puede ir encriptada usando el protocolo POP3S, el cual se conecta usando TSL (Seguridad en capa de transporte) o SSL (capa de conexión segura) sobre el puerto TCP 995.

1.4 IMAP (PROTOCOLO DE ACCESO A MENSAJES DE INTERNET)

Es un protocolo de Internet usado por los clientes de e-mail para obtener los mensajes almacenado en servidor de correo mediante una conexión TCP/IP. Fue diseñado con el propósito de manejar el buzón de correo por múltiples clientes, por lo que generalmente los clientes dejan los mensajes en el servidor hasta que el usuario los borra manualmente.

IMAP permite la manipulación de los buzones de correo electrónico de una forma que es funcionalmente equivalente a carpetas locales y también provee la capacidad de un cliente *offline* volver a sincronizarse con el servidor. (Crispin, 2003) IMAP no especifica una forma de enviar correo; esta función es manejada por SMTP.

Un servidor IMAP típicamente escucha por el puerto 143 y cuando se usa sobre SSL (IMAPS) se asigna el puerto 993. La versión actual de IMAP es la versión 4 definida en el RFC 3501. Soporta los modos "*off-line*" y "*on-line"*

1.5 ALTA DISPONIBILIDAD

Se define alta disponibilidad como la capacidad de mantener el correcto funcionamiento de un servicio constantemente, haciendo uso de una serie de medidas que se aplican con el fin de garantizar dicho propósito. El término que disponibilidad se refiere a la probabilidad de que un servicio esté activo en un momento determinado. La disponibilidad se puede expresar a través del INDICE DE DISPONBILIDAD que es la razón del tiempo que el servicio está disponible sobre el tiempo total de evaluación.

Los fallos de un sistema pueden provocar desde disminución en el rendimiento de la productividad hasta en casos extremos, pérdidas materiales y humanas. Estas fallas son ocasionadas por errores que se presentan en el funcionamiento del software o del hardware del prestador del servicio y las causas de dichos errores se pueden clasificar de la siguiente manera:

1.5.1 Causas físicas

- Desastres naturales (incendios, terremotos, inundaciones)
- Ambiente (temperatura, humedad)
- Fallas materiales
- Fallas de red
- Fallos de fuente de energía.

1.5.2 Causas humanas

- Error de diseño
- Errores operativos
- Daños intencionales

Existen diferentes formas de disminuir las causas de fallas en un servicio tales como la prevención, tolerancia, eliminación y prevención de errores. (Alta disponibilidad, s.f.)

1.6 CLUSTER

En redes se utiliza el término de "CLUSTER" para definir un conjunto de computadores (llamados nodos) que trabajan juntos para desempeñar una tarea específica. Existen cuatro categorías principales de CLUSTERS:

- Almacenamiento
- Alta disponibilidad
- Balanceo de carga
- Alto desempeño

Los CLUSTERS de almacenamiento ofrecen la ventaja de permitir leer y escribir sobre el mismo sistema de archivo simultáneamente aumentando la velocidad de procesamiento, además simplifica la administración de los servidores limitando la instalación de parches y aplicaciones a un solo sistema de archivos.

Un CLUSTER de alta disponibilidad se encarga de mantener la prestación de un servicio constantemente, eliminando causas de fallas y conmutando entre nodos en el caso de que algún servidor falle, evitando que la interrupción sea percibida por el usuario. Para detectar la falla entre los nodos, los servidores monitorean constantemente el estado de los otros componentes del conjunto.

En el balanceo de carga, el CLUSTER envía las peticiones de los servicios de red a través de los nodos para balancear la carga. El número de nodos depende de la cantidad de peticiones que se manejen y la carga se distribuye de tal forma que en el caso de inoperatividad de un nodo el software de balanceo redirecciona las peticiones entre los nodos restantes.

Para el uso de aplicaciones que demanden gran cantidad de recursos operativos se usa el CLUSTER de alto desempeño que permite que los nodos trabajen en paralelo.

1.7 SEGURIDAD EN EL SERVICIO DE CORREO ELECTRÓNICO

Actualmente el correo electrónico es uno de los medios más comunes para la propagación de información, pero lamentablemente este servicio no fue diseñado con las políticas de seguridad pertinentes. En un principio se pensó que la trasmisión se realizaría entre servidores directamente, enviando la información en texto plano, pero con el uso de Internet y con el propósito de personas

mal intencionadas se hizo necesario la implementación de software que asegure la confidencialidad, la autenticidad y la integridad de los datos transmitidos.

En 1991 Phil Zimmerman publicó el software desarrollado por él mismo llamado PGP (*Pretty Good Privacy)* (Bhardwaj, 2011), el cual sirve para cifrar la información y acceder a ella mediante una clave pública, además de permitir firmar digitalmente los documentos para autenticarlos.

Para entender el funcionamiento de este programa, primero es necesario conocer los términos de criptografía y criptografía de llave pública.

La criptografía es la ciencia de usar las matemáticas para encriptar y des-encriptar datos, la finalidad es permitir el envío de información a través de medios inseguros y garantizar que nadie más aparte del destinatario deseado pueda entender el mensaje. En la criptografía convencional también llamada *secret-key* (llave secreta) o cifrado simétrico, se usa una única llave para cifrar y descifrar la información.

Un método más seguro para cifrar datos es la criptográfica de llave pública (*public key cryptography*), el cual es un esquema asimétrico que usa un par de llaves para el cifrado: una llave pública, que encripta los datos, y una correspondiente llave privada o llave secreta que descifra la información.

El programa PGP combina las mejores características de los dos métodos. Cuando un usuario usa PGP para encriptar datos, primero comprime el texto plano, luego el programa crea una llave secreta que solo se usa una vez, esta llave es un número aleatorio generado de acuerdo al movimiento del mouse y de las teclas que se utilicen. Una vez se complete este proceso, los datos son encriptados con la llave pública del destinatario y luego se envía el mensaje. Para descifrar los datos el destinatario usa su clave privada para recuperar la llave temporal, la cual PGP usa para des-encriptar el mensaje. (How PGP works, s.f.)

1.8 ATAQUES A TRAVES DEL SERVICIO DE CORREO ELECTRÓNICO

1.8.1 Correos electrónicos no deseados (Spam)

Se define SPAM o correo basura a todos los mensajes no solicitados, no deseados o de remitente desconocido que diariamente se envían a través del correo electrónico y otros medios de comunicación. Se estima que en 60% de los correos enviados son no solicitados, esto representa una cantidad de datos enorme que ocupa los recursos de la red sin proporcionar ningún beneficio.

Algunas de las consecuencias de recibir SPAM en una empresa son:

- Reducción en la productividad de los empleados, al invertir tiempo leyendo información de publicidad que es lo que generalmente contienen este tipo de mensajes.
- Reducción de capacidad de almacenamiento en el servidor de correo y en los equipos cliente.
- Posible infección con *malware* que en algunos casos son distribuidos por medio de este tipo de correos.

Generalmente los mensajes SPAM son enviados masivamente, utilizando listas de direcciones de correo que pueden ser robadas, compradas o recolectadas en la web. (Spam, s.f.)

1.8.2 Phishing

Este mecanismo de estafa busca robar información personal y con ella tener acceso a servicios privados, generalmente bancarios, para obtener beneficio propio. A través del envío de correos electrónicos masivos, los intrusos dan instrucciones para que los usuarios entren a direcciones de páginas web falsas (copia de sitios de instituciones financieras, tiendas, etc.), en donde se solicita realizar cambios de la información personal. Los datos como los nombres de usuario y contraseñas son el objetivo de los intrusos. (Fuentes Serrano & Calderón Hernández, s.f.)

1.9 MAIL RELAY

Este problema se presenta cuando individuos maliciosos hacen uso de los recursos de un servidor de correo para retransmitir un mensaje a cientos de direcciones, además de reducir el desempeño del equipo, el servidor puede ser atacado por numerosos mensajes de error de envío y lo peor, el servidor puede ser añadido a listas negras por parecer que está difundiendo de forma masiva mensajes de *SPAM*. (Securing SMTP mail servers, 2010)

1.10 DRBD (DISTRIBUTED REPLICATED BLOCK DEVICE)

Es una herramienta de software que tiene como propósito facilitar la duplicación de la información entre equipos eliminando la dependencia entre ellos. La replicación se puede hacer de la información contenida en discos duros, particiones, volúmenes lógicos etc. Los fundamentos del funcionamiento de esta herramienta son:

- La replicación ocurre constantemente mientras los datos de las aplicaciones se modifican. (en tiempo real).
- Para las aplicaciones el almacenamiento de los datos es transparente, a pesar de ser guardados en múltiples dispositivos.

DRBD soporta tres diferentes tipos de modos de duplicación, lo cual permite obtener tres grados de sincronismo en la replicación.

Protocolo A: Protocolo de replicación asíncrona. En el nodo primario se considera que una operación está completa tan pronto como el disco escribe el final de la tarea y el paquete de replicación es ubicado en el buffer de envío local. En este modo, en el caso de una falla en el nodo primario, las últimas actualizaciones realizadas serán información perdida. Este protocolo se usa comúnmente en escenarios donde hay distancias considerables entre los equipos.

Protocolo B: Protocolo de replicación de memoria sincronizada (semi-sincronizado). Se considera que una operación está terminada tan pronto como la escritura en el disco haya ocurrido y cuando el paquete de replicación ha alcanzado el nodo par. En el caso de presentar fallas en los dos nodos simultáneamente, los datos más recientes se perderán.

Protocolo C: Protocolo de replicación síncrona. Solamente se considerará una operación como completa, cuando los discos del equipo local y del equipo remoto confirmen que esto ha sucedido.

Como resultado, se garantiza que no haya pérdida de datos con la caída de un nodo. (Hellman, Haas, Reisner, & Ellenberg, 2011)

1.11 ZIMBRA COLLABORATION 8.6 OPEN SOURCE EDITION

Zimbra es una aplicación diseñada para el envío de mensajes que incluye herramientas como el email, lista de contactos, calendario, tareas, entre otras. La arquitectura de esta solución consiste en interfaces de usuario y componentes de servidor que pueden ser configurados en un solo equipo o en múltiples nodos para obtener alta disponibilidad e incrementar la escalabilidad.

La arquitectura se caracteriza por:

- Integración Open Source. Uso de software libre como Linux, MariaDB, Postfix entre otros.
- Uso de los protocolos de estándar industrial. SMTP, LMTP. SOAP. XML, IMAP, POP.
- Escalabilidad horizontal. Es posible integrar en un sistema múltiples servidores.
- Buscador en la interface de usuario. Permite tener fácil acceso a las herramientas de Zimbra.
- Buscador en la consola de administración.

Algunos de los principales componentes de estructura del software son:

1. Jetty, es el servidor de aplicación web con el que Zimbra funciona.
2. Postfix, es un MTA (Mail Tranfer Agent) libre que direcciona los mensajes al servidor apropiado.
3. MariaDB, software de manejo para las bases de datos.[1]
4. Componentes libres de anti-spam y anti-virus.
5. LibreOffice, utilizado para la vista previa de los documentos.

Manejo de la información en la aplicación Zimbra

El servidor de correo proporciona a cada usuario una cuenta la cual es configurada en un *mailbox server*, cada cuenta está asociada a un *mailbox* que contiene información como los mensajes, contactos, calendario y archivos adjuntos. Cada *mailbox server* tiene su propio almacenamiento para los mensajes, datos e información de indexación para los *mailbox* que contiene.

El almacenamiento para mensajes utiliza el formato MIME (Multipurpose Internet Mail Extensions) para el cuerpo de los mensajes y para los archivos adjuntos.

El almacenamiento de datos se maneja en una base de datos MariaDB, donde se almacenan los números de identificación de cada *mailbox.*

La indexación y la tecnología de búsqueda se suministrada a través de Apache Lucene[2]. Cada mensaje y archivo adjunto es indexado automáticamente cuando llega, de esta forma esta información es asociada a cada cuenta. (Zimbra, Inc, 2014)

[1] Profundización en https://mariadb.com/kb/en/mariadb/what-is-mariadb/

[2] Profundización en https://lucene.apache.org/core/

Para entender mejor el funcionamiento del software, a continuación de presenta una breve explicación de las principales funciones de Zimbra y de los objetos en los que se basa el manejo de las cuentas de correo:

Objetos de Zimbra Collaboration

Para entender mejor el funcionamiento del software de correo es necesario conocer los objetos que maneja.

- Cuentas: representan una cuenta del servidor Zimbra. Existen dos tipos, de administrador y de usuario, la primera tiene acceso a la consola de administración y desde allí se puede modificar todos los parámetros de configuración. La cuenta de usuario tiene ciertos atributos denominados preferencias.
- Clase de Servicio (COS): las COS controlan las características, restricciones y privilegios que va a manejar cada cuenta. Al crear una cuenta, es necesario asociarla a una clase de servicio previamente establecida.
- Dominios: representa el dominio del correo electrónico.
- Listas de distribución: son usadas para agrupar las cuentas bajo un mismo nombre, de tal forma que solo sea necesario enviar un correo al nombre de la lista para que el mensaje llegue a todas las cuentas que pertenezcan al grupo.
- Alias: representa un sobrenombre que se puede aplicar para las cuentas y para las listas de distribución.
- Zimlets: son herramientas adicionales que se pueden instalar en Zimbra para agregar funciones.

Las principales características que maneja el software son:

- Con el correo la cual habilita la opción de enviar y recibir mensajes.
- Con la herramienta de agenda (*Calendar*), el usuario puede programar citas, incluso con usuarios de correo de otros servidores y puede configurar recordatorios.
- Usando la opción de tareas el usuario puede llevar un control del desarrollo de obligaciones propias.
- El maletín es una herramienta muy útil para el manejo de información, por esta razón está habilitada para todos los usuarios. Con el maletín se almacenan archivos directamente en el servidor, de tal forma que siempre que tenga acceso a su cuenta de correo, el usuario puede hacer uso de la información que ahí está almacenada. Además de organizar los archivos por carpetas, el usuario puede compartir la información con otro usuario que maneje una cuenta de correo en el mismo servidor, definiendo que permisos le otorga a cada persona con quien comparte los datos.

1.12 SPAMASSASSIN

Es un filtro de correo electrónico que identifica el spam (Correo no deseado). Usa una diversa variedad de pruebas para reconocer correos masivos indeseados. Estas pruebas son aplicadas al encabezado y contenido del mensaje, para clasificarlo usando métodos estadísticos y heurísticos como análisis del texto, filtrado Bayesiano, listas de bloqueo DNS y base de datos colaborativos. (Apache Foundation, 2016)

Sin embargo, SpamAssassin no elimina el spam, mueve el spam a buzones separados ni envía rebotes cuando se recibe spam. Estas son funciones de un enrutador de correo y SpamAssassin no lo es.

Spam Assassin incluye un complemento para reportar mensajes de spam automáticamente a las bases de datos colaborativas, cuenta con una amplia comunidad y diferencia exitosamente típicamente entre el 95% y 100% de los casos, dependiendo de qué tipo de correos reciba y el entrenamiento de sus filtros Bayesianos. Algunos casos específicos han producido alrededor de 1.5% de falsos negativos (spam que no se detectó) y 0.06% de falsos positivos (correo marcado como spam)

1.13 CLAMAV

Es un software antivirus gratis y de código abierto que detecta varios tipos de software malicioso. Es usado principalmente como un escáner de virus de correo en lado del servidor. Es el antivirus integrado con Zimbra por defecto.

1.14 REGISTROS MX

En los sistemas de nombre de dominio (DNS, *Domain Name System*) los registros MX son un tipo de recurso, donde se especifica el servidor de correo responsable de aceptar los mensajes del dominio receptor e indicar la preferencia para priorizar los servidores de recepción si hay múltiples disponibles. El conjunto de registros mx de un dominio especifican como un correo electrónico debe ser encaminado con el SMTP.

Cuando un mensaje de email es enviado por Internet, el agente de transferencia de correo consulta el sistema de nombres de dominio por el registro mx del dominio de cada destinatario. Esta consulta devuelve una lista de nombres de servidores de intercambio de correo (*Mail eXchangers*), los cuales aceptan los correos para esos dominios. Luego el agente intenta establecer una conexión SMTP.

Un spammer es quien envía correos no deseados de forma masiva y constante. Estos frecuentemente envían correo al servidor de menor preferencia, esto con la intención de evitar filtros anti-spam, que pueden estar ejecutándose en el servidor principal. En algunas ocasiones los servidores de respaldo pueden estar usando diferente software anti-spam lo que le permite ocultar la dirección IP del servidor primario.

Algunas veces, los *spammers* solo atacan los servidores con la mejor preferencia, y no buscan el siguiente servidor cuando no está disponible el principal. Una técnica llamada *nolisting,* aprovecha este comportamiento al crear un registro MX de un servidor primario no existente.

1.15 RBL (REALTIME BLACKHOLE LIST)

Las RBL son listas de direcciones IP y dominios los cuales se niegan a parar la difusión de mensajes de correo con *SPAM***.** Los prestadores de servicio y las compañías en general consultan dichas listas para conocer los emisores a bloquear, el bloqueo de la mayoría del tráfico sucede en la fase de conexión SMTP.

2 INFRAESTRUCTURA DEL PROYECTO

El desarrollo del trabajo de aplicación, se basa en la implementación de un cluster de alta disponibilidad, compuesto por dos nodos suministrados por Offimedicas S.A. Cada nodo es un servidor Dell poweredge T110II con procesador Intel Xeon de 4 núcleos, 8 GB de memoria ram, 4TB de capacidad de disco duro sata de 7200rpm, a los cuales se les instala CentOS 7, como sistema operativo.

Los servidores cuentan con fuentes eléctricas de respaldo, que consisten en una planta eléctrica con motor de combustión interna, que alimenta a dos UPS, conectadas a una PDU, que provee la alimentación ininterrumpida de los servidores. Así, Offimedicas mitiga los riegos de interrupción del servicio por fallas en el flujo eléctrico.

Cada nodo tiene dos interfaces de red, una para comunicarse con la red local, por donde los usuarios acceden al servicio de correo, y la otra para mantener la sincronización entre los discos del cluster. Cada interface de red está configurada con dirección IP estática y adicionalmente, el cluster maneja una dirección IP virtual que está asociada al nodo que corre el servicio de correo.

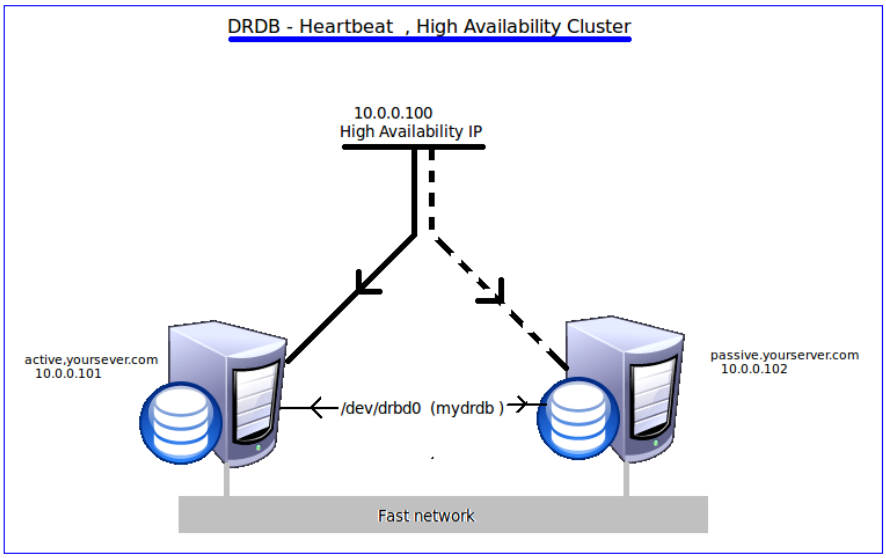

Ilustración 1. Infraestructura del cluster[3]

Zimbra almacena los datos de configuración, los mensajes de correo, los archivos del maletín y demás datos de la aplicación en el directorio "/opt/zimbra/", por lo cual debe tener la mayor cantidad de almacenamiento posible. Esta ubicación se replica en tiempo real mediante DRBD, quien copia las particiones en las cuales se montó este directorio.

[3] Imagen tomada de http://www.sherin.co.in/wp-content/uploads/2010/06/drdb_hearbeat.png

3 INSTALACIÓN Y CONFIGURACIÓN

3.1 INSTALACIÓN DEL SISTEMA OPERATIVO CENTOS 7

La mayoría de los servicios de red en OFFIMEDICAS S.A. están instalados en equipos que corren el sistema operativo CentOS, por esta razón se selecciona la última versión de éste para la implementación del proyecto. La instalación del sistema operativo en los equipos servidor se realizó siguiendo los siguientes pasos[4]:

1. Seleccionar idioma “English (United States)” para ser utilizado en el proceso de instalación.

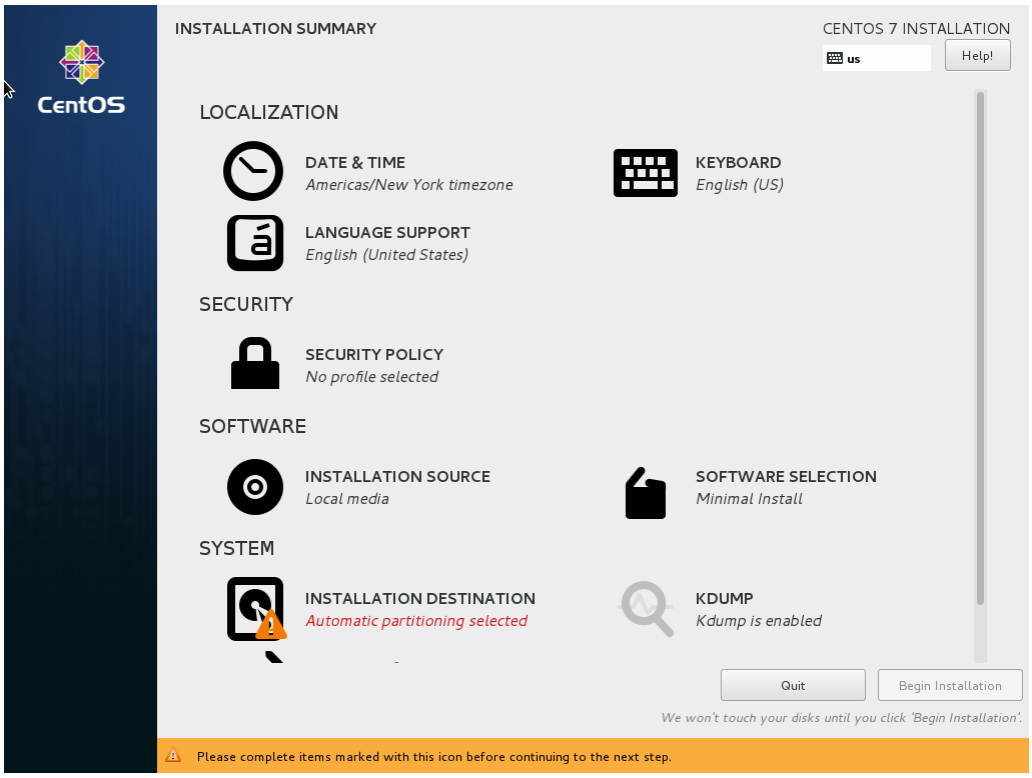

Ilustración 2 Opciones de configuración

2. Las opciones de configuración para el proceso de instalación se presentan en la Ilustración 2. La selección del software a instalar se deja con el valor por defecto (instalación mínima), el cual cumple con los requisitos necesarios para la realización del objetivo, por lo tanto, no se modifica la sección “SOFTWARE SELECTION”.

[4] Las imágenes de la instalación del sistema operativo son una recreación de la instalación real hecha en VirtualBox para facilitar la toma.

3. Debido a que el servidor se encuentra protegido por un firewall empresarial, no es necesario ejecutar un firewall adicional en éste. Por ésta razón, en el módulo de seguridad no se hace selección de ningún perfil y así se evita interferir con el funcionamiento de las aplicaciones necesarias para el desarrollo del proyecto.

4. En “NETWORK & HOST NAME” se establecen los parámetros de conexión del equipo en donde se realiza la asignación manual de IP, para evitar que al utilizar el protocolo DHCP, las direcciones de los servidores cambien e interfieran con la sincronización de datos. (Ver Ilustración 3). En el servidor Mercurio se usa la dirección IP 10.131.210.51 con máscara 24 para la interfaz conectada a la red de Offimedicas S.A. y la dirección IP 192.168.99.1 con máscara 30 para la conexión directa a Hermes. En “Host Name:” se escribe el FQDN que éste caso es mercurio.offimedicas.com.

   En el servidor Hermes se configura la dirección IP 10.131.210.52 con máscara 24 en la interfaz conectada a la red de Offimedicas S.A. y la dirección IP 192.168.99.2 con máscara 30 para la conexión directa a Mercurio. En “Host name:” se escribe hermes.offimedicas.com para este servidor.

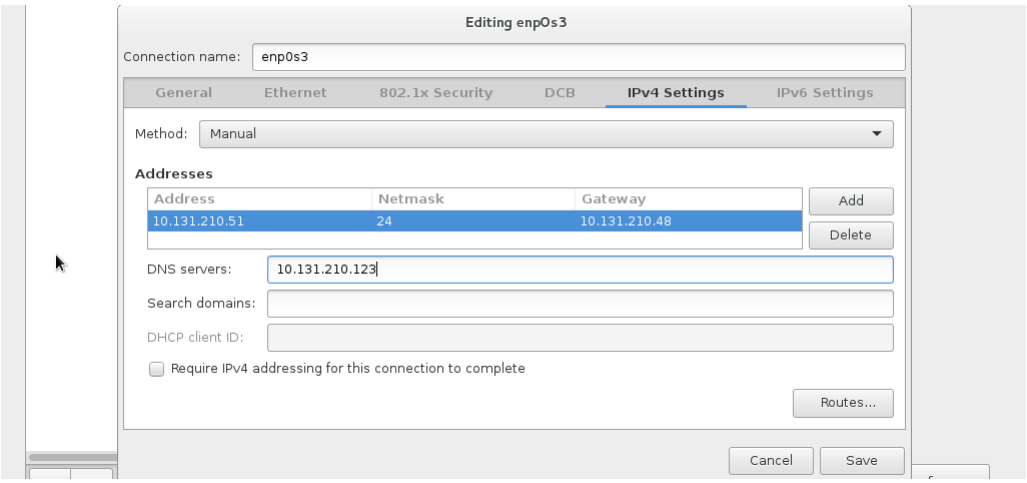

Ilustración 3 Asignación de dirección IP

Como se muestra en Ilustración 4 la forma de asignar el nombre al servidor y de encender la interface de conexión, es llenando el campo de nombre de host y activando el botón de encendido respectivamente.

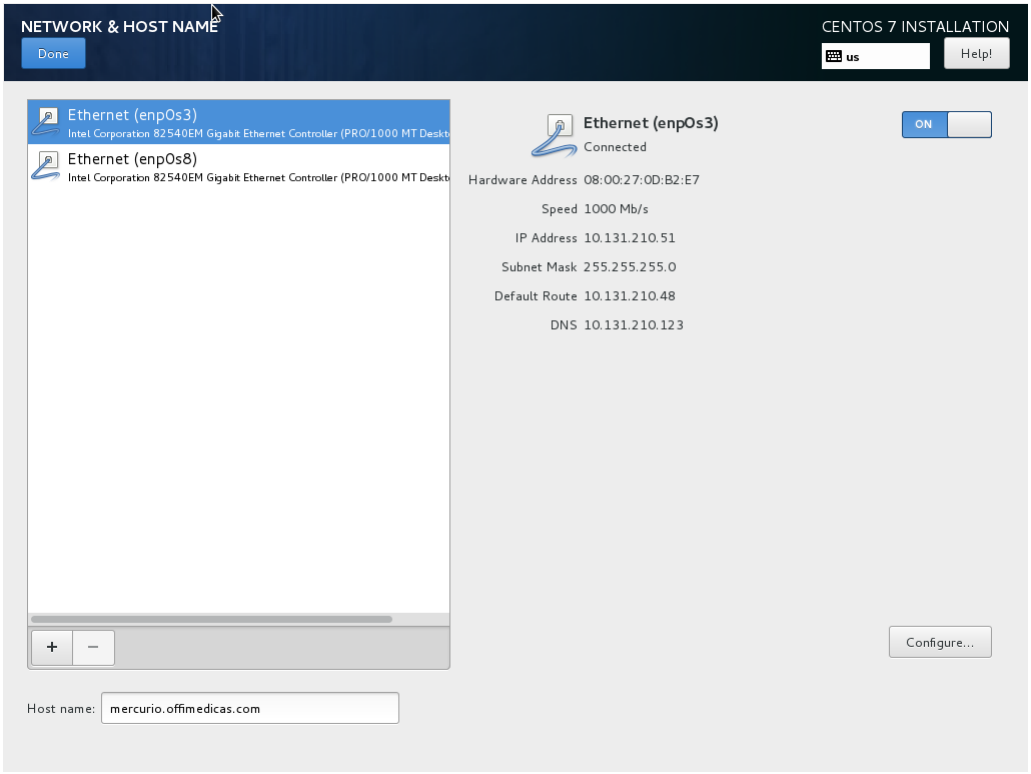

Ilustración 4 Definición nombre del equipo

5. Configurar el disco en la opción "INSTALLATION DESTINATION" donde se define manualmente el manejo de las particiones en la opción "I will configure partitioning." para asegurar que el espacio disponible en el disco sea utilizado de la forma más conveniente. Para facilitar la creación de las particiones, se presiona sobre "Click here to create them atumatically" (ver Ilustración 5). Se debe eliminar las particiones diferentes a "*/boot", "/" y "swap" y* reducir la capacidad de la partición root "*/" hasta el mínimo deseado*.

   Es necesario reclamar el espacio libre dentro del volumen lógico, para luego poder asignarle una partición que DRBD usará para la replicación. Esto se logra modificando la politca de tamaño como se ve en Ilustración 6. El resultado del particionamiento se ve en Ilustración 7.

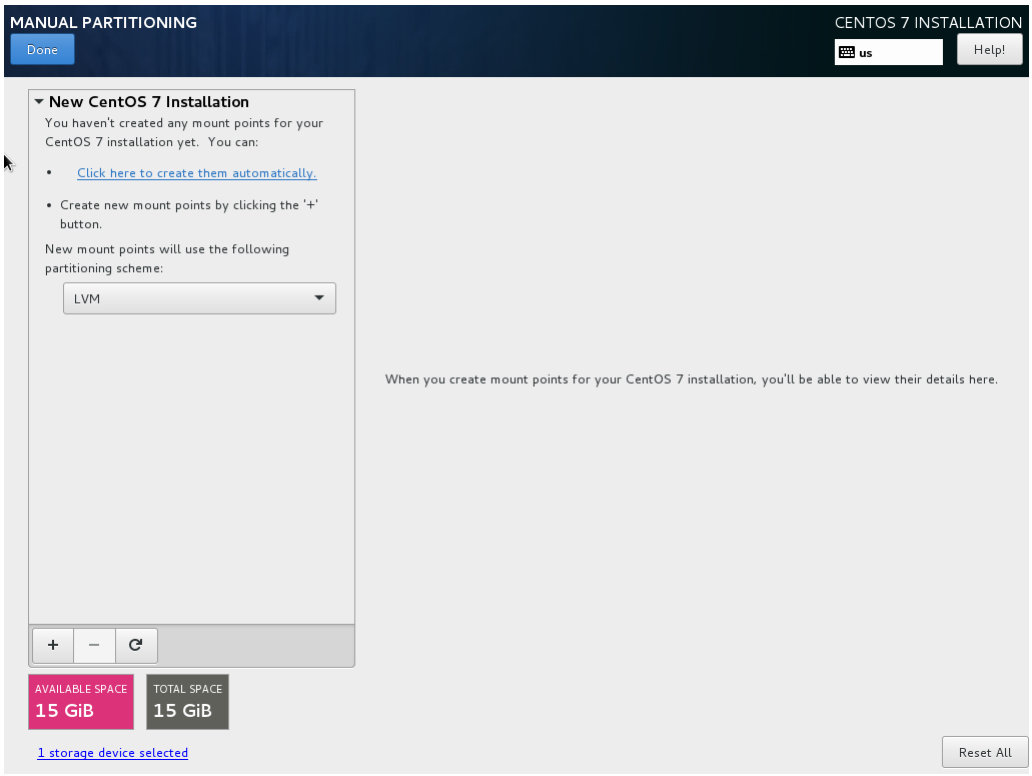

Ilustración 5. Destino de instalación.

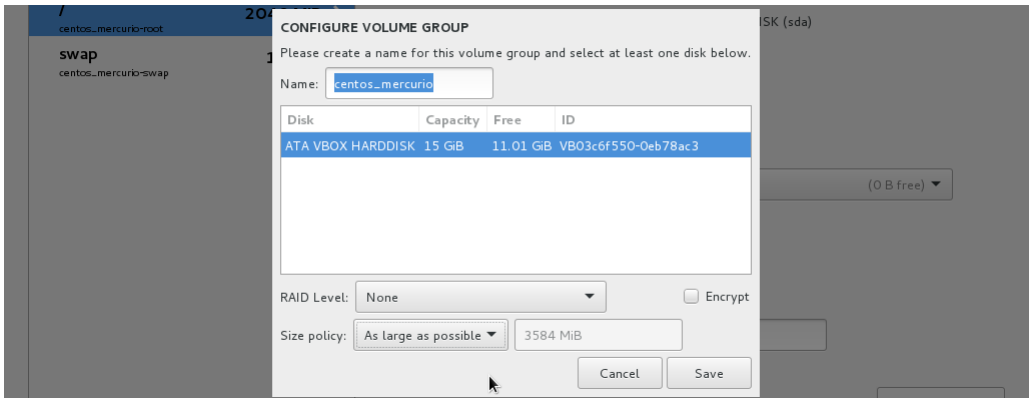

Ilustración 6. Configuración del volumen lógico

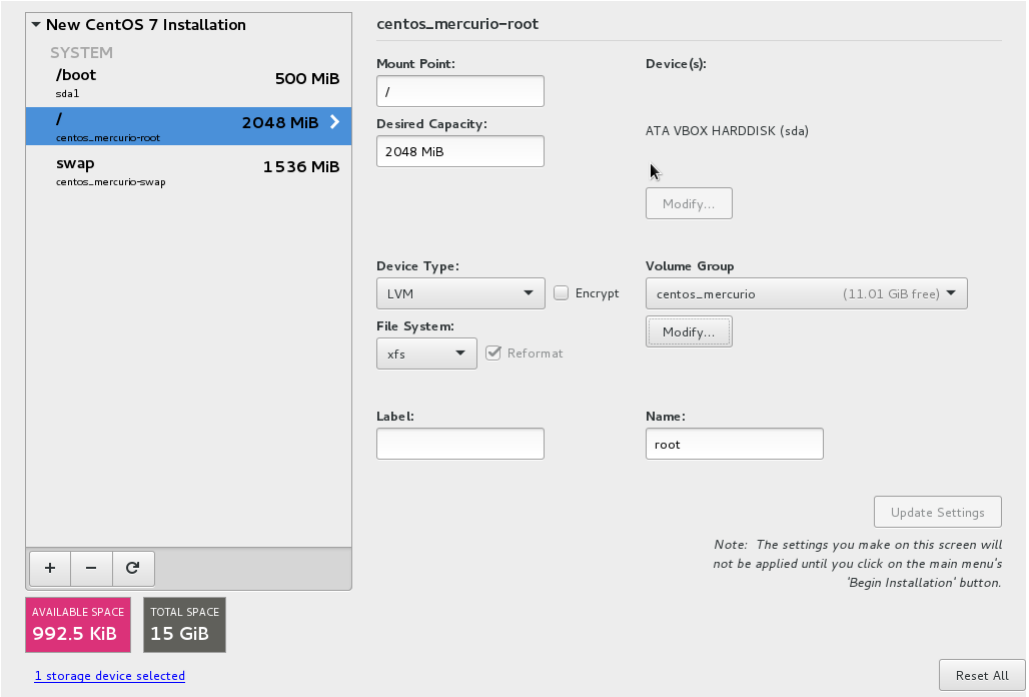

Ilustración 7. Configuración partición root

6. En la opción de “DATE & TIME” se escoge la zona horaria de acuerdo al lugar en donde esté el servidor y se activa la opción de “Network Time” con la cual el equipo se sincroniza con un servidor NTP.

7. En “KEYBOARD” se selecciona el teclado apropiado, y se hace clic sobre “Begin Installation”. De esta manera, comienza la instalación del sistema operativo con una barra de progreso y un ícono de selección para establecer la contraseña del usuario “root”, que es el usuario con todos permisos para hacer modificaciones en el sistema.

3.2 REQUISITOS DEL SISTEMA PARA LA IMPLEMENTACIÓN DE ALTA DISPONIBILIDAD

Para asegurar la correcta instalación y funcionamiento de las herramientas que se utilizan en el proyecto (servicio de correo, duplicación de datos y alta disponibilidad) se deben cumplir con ciertos requisitos que se describen a continuación:

- Deshabilitar SELinux[5]; para ello es necesario modificar el archivo de configuración ubicado en /etc/selinux/config, cambiando la línea que dice:

  ```
  SELINUX=enforcing
  ```
  por
  ```
  SELINUX=disabled
  ```

  Hacer esto en cada uno de los nodos y reiniciar los equipos.

- Deshabilitar Sendmail y Postfix en cada uno de los nodos
  ```
  # systemctl disable sendmail.service
  # systemctl disable postfix.service
  ```

- Actualizar el sistema operativo
  ```
  #yum update -y
  ```

- Instalar los siguientes paquetes de software: NPTL, Netcat (nc), sudo, libidn, gmp y libaio.
  ```
  #yum install nc sudo libidn gmp libaio sysstat perl-core
  sqlite unzip
  ```

- Importar el paquete del repositorio ELREPO y habilitarlo.
  ```
  # rpm --import https://www.elrepo.org/RPM-GPG-KEY-elrepo.org
  # rpm -Uvh http://www.elrepo.org/elrepo-release-7.0-
  2.el7.elrepo.noarch.rpm
  ```

- Instalar el módulo kernel de DRBD y sus utilidades.
  ```
  # yum install -y kmod-drbd84 drbd84-utils
  ```

- Instalar Pacemaker y sus herramientas

  ```
  # yum install -y pacemaker pcs psmisc policycoreutils-python
  ```

[5] Si se tiene firewall instalado, es mejor deshabilitarlo si se tiene en una red protegida. De lo contrario se deben abrir el puerto 7789 para DRBD en las interfaces de red que comunican los servidores. Los puertos 80 para acceso web, 465 para el protocolo SMPTS, 25 para SMTP, 993 para IMAPS, 143 para IMAP, 995 para POP3S, 110 para POP3, 7025 para LMTP, 443 para accesos web cifrado y 7071 para la consola de administración web, deben abrirse en la interfaz de red que se comunica con los clientes.

3.3 CONFIGURACIÓN DE NOMBRES DE DOMINIO

Una de las actividades a realizar es configurar los registros MX y los registros A en los servidores de nombres de dominio, con el objetivo de que los demás servidores de correo puedan determinar hacia dónde van los mensajes del dominio offimedicas.com.

3.3.1 Registros A

En los registros A se especifica la relación que hay entre la dirección IP publica donde se encuentra alojado el servidor de correo y la URL que maneja el servicio como se muestra en la Ilustración 8.

Registro personalizado				
mercurio.offimedicas.com	86400	IN	A	181.48.83.85

Ilustración 8 Registros A para servidor de correo

3.3.2 Registros mx

En los registros mx se indica a donde se deben enviar los correos, en la Ilustración 9 se muestra que todos los mensajes que llegan con el dominio @offimedicas.com se envían a mercurio.offimedicas.com, el cual está definido en los registros A con la dirección IP de servidor.

Registro MX personalizado				
offimedicas.com	86400	IN	MX	1 mercurio.offimedicas.com

Ilustración 9 registros MX

3.3.3 Pruebas de los registros

Con el comando nslookup podemos revisar que esté funcionando correctamente el nombre de dominio. En la Ilustración 10 se observa que el registro mx para el dominio offimedicas.com es mercurio.offimedicas.com y tiene una preferencia de 1. Adicionalmente se verifica que la dirección IP para mercurio.offimedicas.com es la dirección IP pública del servidor de correo.

3.3.4 Configuración dominios internos con BIND

En el servidor local DNS de la empresa, también se deben agregar los registros necesarios para que al momento de instalar Zimbra este lo reconozca como un servidor de correo. En la zona offimedicas.com se agregaron los registros A y los MX de la siguiente forma:

```
ns1.offimedicas.com.          IN      A       10.131.210.123
correo.offimedicas.com.       IN      A       10.131.210.50
mercurio.offimedicas.com.     IN      A       10.131.210.51
hermes.offimedicas.com.       IN      A       10.131.210.52
offimedicas.com.              IN      MX 10 correo.offimedicas.com.
```

Donde el dominio correo.offimedicas.com está apuntando a la dirección virtual que cambiara de equipo según la disponibilidad. De igual manera se configura la zona inversa:

```
; PTR Records
123     IN      PTR     ns1.offimedicas.com       ; 10.131.210.123
50      IN      PTR     correo.offimedicas.com    ; 10.131.210.50
51      IN      PTR     mercurio.offimedicas.com  ; 10.131.210.51
52      IN      PTR     hermes.offimedicas.com    ; 10.131.210.52
```

```
C:\Users\saba_>nslookup mercurio.offimedicas.com
Servidor:  BlueTiger.Home
Address:  10.131.210.48

Respuesta no autoritativa:
Nombre:  mercurio.offimedicas.com
Address:  181.48.83.85


C:\Users\saba_>nslookup
Servidor predeterminado:  BlueTiger.Home
Address:  10.131.210.48

> set type=mx
> offimedicas.com
Servidor:  BlueTiger.Home
Address:  10.131.210.48

Respuesta no autoritativa:
offimedicas.com MX preference = 2, mail exchanger = hermes.offimedicas.com
offimedicas.com MX preference = 1, mail exchanger = mercurio.offimedicas.com
>
```

Ilustración 10. Prueba de nombre de dominio

3.3.5 Prueba DNS interno

En el equipo con el DNS configurado con la dirección IP 10.131.210.123 probaron los registros agregados

```
[root@mercurio ~]# nslookup correo.offimedicas.com
Server:         10.131.210.123
Address:        10.131.210.123#53

Name:   correo.offimedicas.com
Address: 10.131.210.50

[root@mercurio ~]# nslookup
> set type=mx
> offimedicas.com
Server:         10.131.210.123
Address:        10.131.210.123#53

offimedicas.com   mail exchanger = 10 correo.offimedicas.com.
```

4 CONFIGURACIÓN DEL CLUSTER

4.1 HABILITAR LOS DEMONIOS DE PCS Y CONFIGURAR COROSYNC[6]

Para poder configurar el cluster es necesario iniciar el demonio *pcs* (*pcsd*). Este demonio trabaja con el comando pcs desde la interfaz de línea de comandos para administrar la sincronización de los archivos de *corosync* en todos los nodos del cluster. Se inicia y habilita con el arranque del equipo respectivamente con los siguientes comandos:

```
# systemctl start pcsd.service
# systemctl enable pcsd.service
```

También se debe habilitar al arranque del sistema Pacemaker y Corosync

```
# systemctl enable corosync.service
# systemctl enable pacemaker.service
```

El usuario *hacluster,* que es creado al instalar los paquetes de *pacemaker*, no cuenta con contraseña y se le debe asignar una para poder sincronizar los archivos de *corosync* o iniciar y detener el cluster en otros nodos. Para asignar la contraseña en cada nodo se usan el siguiente comando. Se debe hacer en ambos nodos y asignar la misma contraseña.

```
#passwd hacluster
```

Es necesario autenticarse para poder ejecutar los comandos en los nodos del cluster, se usa el comando `pcs cluster auth` para autenticarse con el usuario *hacluster.* Solamente debe hacerse en uno de los nodos.

```
[root@mercurio etc]# pcs cluster auth mercurio hermes
Username: hacluster
Password:
mercurio: Authorized
hermes: Authorized
```

Luego, en el mismo nodo, se crea y sincroniza el archivo de configuración de corosync.

```
[root@mercurio ~]# pcs cluster setup --name cloffimedicas mercurio
Hermes
```

[6] Corosync y Pacemaker son un conjunto de desarrollos destinados a aumentar la disponibilidad de servicios. Para más información ver: http://corosync.github.io/corosync/ y http://clusterlabs.org/

4.2 CREACIÓN DE UN CLUSTER ACTIVO/PASIVO CON EL RECURSO DE DIRECCIÓN IP VIRTUAL.

Para el correcto funcionamiento del servicio de correo es necesario crear una dirección IP única que el cluster pueda asignar a uno de los nodos. De esta forma, los usuarios pueden seguir accediendo al servicio sin importar en cual nodo se está ejecutando. Para esto se asignará un recurso con dirección IP 10.131.210.50, nombre ClusterIP y se le dirá al cluster que revise si está activa cada 30 segundos. Con el siguiente comando se crea el recurso y se establece la frecuencia de revisión mencionada con el script IPaddr2.

```
[root@mercurio ~]# pcs resource create ClusterIP ocf:heartbeat:IPaddr2 ip=10.131.210.50 cidr_netmask=24 op monitor interval=30s
```

Con este comando se le indica a *Pacemaker* que el recurso se encuentra en ***ocf****, que es el estándar al que pertenece el script. La parte del* ***heartbeat***, que es específico de *ocf,* indica en cual *namespace* se encuentra el script IPaddr2.

Con el comando `pcs property set stonith-enabled=false` se deshabilita la función de "*fencing"* que consiste en aislar a un nodo que se esté comportando de forma errática. De esta forma la decisión de en qué nodo se ejecuta el servicio queda en manos de pacemaker.

Con el comando pcs status podemos listar los servicios activos

```
[root@mercurio etc]# pcs status
Cluster name: cloffimedicas
Last updated: Sat May 21 16:59:29 2016          Last change: Sat May 21 16:59:24 2016 by root via cibadmin on mercurio
Stack: corosync
Current DC: mercurio (version 1.1.13-10.el7_2.2-44eb2dd) - partition with quorum
2 nodes and 1 resource configured

Online: [ hermes mercurio ]

Full list of resources:

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started hermes


PCSD Status:
  mercurio: Online
  hermes: Online


Daemon Status:
```

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

Mediante el comando `ip addr` podemos verificar que el sistema tenga su dirección IP virtual asignada:

```
[root@hermes ~]# ip addr
    link/ether 08:00:27:fe:90:ce brd ff:ff:ff:ff:ff:ff
    inet 10.131.210.50/24 brd 10.131.210.255 scope global secondary
enp0s3
       valid_lft forever preferred_lft forever
```

Esta dirección IP es la que está configurada como la dirección de correo electrónico, en los registros A y los registros MX del servidor DNS.

5 REPLICACIÓN DE DATOS USANDO DRBD

Para la configuración de DRBD, es necesario contar con la instalación de los paquetes descritos en la sección 3.2. Cumplidos estos requerimientos se continúa con los siguientes pasos:

5.1 VOLUMEN LÓGICO PARA DRBD

Es necesario designar un volumen lógico para DRBD en cada uno de los nodos, teniendo en cuenta que para ambos equipos el tamaño del volumen lógico debe ser igual.

```
[root@mercurio etc]# lvcreate --name drbd-zimbra --size 11G
centos_mercurio

[root@hermes etc]# lvcreate --name drbd-zimbra --size 11G
centos_hermes
```

5.2 CONFIGURACIÓN DE DRBD

Se crea el siguiente archivo en cada uno de los nodos del cluster.

```
# cat <<END >/etc/drbd.d/zimbradata.res
resource zimbradata {
protocol C;
meta-disk internal;
device /dev/drbd1;
syncer {
verify-alg sha1;
}
net {
allow-two-primaries;
}
on mercurio.offimedicas.com {
disk /dev/centos_mercurio/drbd-zimbra;
address 192.168.99.1:7789;
}
on hermes.offimedicas.com {
disk /dev/centos_hermes/drbd-zimbra;
address 192.168.99.2:7789;
}
}
END
```

- Donde se crea el recurso de DRBD llamado zimbradata.
- Se configura la replicación completamente síncrona al elegir Protocol C.
- Se involucran todos los nodos que conforman el cluster.
- Se define que el puerto de conexión a la red es 7789.

5.3 INICIALIZACIÓN DE DRBD

1. Crear la metadata local para el recurso DRBD.

   ```
   [root@mercurio drbd.d]# drbdadm create-md zimbradata
   ```

2. Luego se asegura que el módulo kernel de DRBD este cargado, invocándolo nuevamente.

   ```
   #modprobe drbd
   ```

3. Finalmente encender el recurso DRBD creado. Estos comandos se aplican en todos los nodos.

   ```
   #drbdadm up zimbradata
   ```

   Para verificar el estado de DRBD se utiliza el siguiente comando:

   ```
   [root@hermes etc]# cat /proc/drbd

   version: 8.4.7-1 (api:1/proto:86-101)

   GIT-hash: 3a6a769340ef93b1ba2792c6461250790795db49 build by
   phil@Build64R7, 2016-01-12 14:29:40

    1: cs:Connected ro:Secondary/Secondary
   ds:Inconsistent/Inconsistent C r-----

       ns:0 nr:0 dw:0 dr:0 al:8 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1
   wo:f oos:11533948
   ```

4. Establecer el rol de nodo primario en el cluster.

   ```
   # drbdadm primary --force zimbradata
   ```

En este punto la tarea de sincronización de datos está en proceso, cuando ésta finaliza y se verifica el estado de la sincronización, se obtiene una respuesta en donde se visualiza los roles de cada nodo y se confirma la conexión entre ellos.

```
# cat /proc/drbd
version: 8.4.6 (api:1/proto:86-101)
GIT-hash: 833d830e0152d1e457fa7856e71e11248ccf3f70 build by
phil@Build64R7, 2015-04-10
05:13:52
1: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C
r-----
ns:1048508 nr:0 dw:0 dr:1049420 al:0 bm:0 lo:0 pe:0 ua:0 ap:0
ep:1 wo:f oos:0
```

5.4 MANEJO DE DRBD CON PACEMAKER

Pacemaker se debe encargar de manejar el servicio de DRBD, montando el disco en el equipo que va a estar corriendo el recurso de la dirección IP virtual, así en caso de que un equipo falle, Pacemaker montará el disco con la información replicada en el otro servidor.

Lo primero es crearle un sistema de ficheros al disco, para que se pueda utilizar en el sistema operativo. El sistema de ficheros de los servidores CentOS es "*xfs*", por lo tanto, se utiliza el siguiente comando:

```
[root@mercurio drbd.d]# mkfs.xfs /dev/drbd1
meta-data=/dev/drbd1             isize=256    agcount=4, agsize=720872
blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=0        finobt=0
data     =                       bsize=4096   blocks=2883487, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=0
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

Ahora se crean los recursos que manejen el disco. Para crearlos, se graban las configuraciones de los recursos en un archivo llamado drbd_cfg y luego se cargan a Pacemaker.

Se crea el archivo drbd_cfg con la configuración actual:

```
[root@mercurio ~]# pcs cluster cib drbd_cfg
```

Para hacer cambios en el archivo se usa el comando "pcs -f". Se crea un recurso de Cluster para el dispositivo DRBD, y agrega un recurso clonado para permitir correr éste recurso en ambos nodos al mismo tiempo.

```
[root@mercurio ~]# pcs -f drbd_cfg resource create mailData
ocf:linbit:drbd drbd_resource=zimbradata op monitor interval=60s

[root@mercurio ~]# pcs -f drbd_cfg resource master mailDataClone
mailData master-max=1 master-node-max=1 clone-max=2 clone-node-
max=1 notify=true
```

Para observar los cambios hechos en la configuración:

```
[root@mercurio drbd.d]# pcs -f drbd_cfg resource show

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started hermes
 Master/Slave Set: mailDataClone [mailData]
```

```
     Stopped: [ hermes mercurio ]
```

Para subir los cambios:

```
[root@pcmk-1 ~]# pcs cluster cib-push drbd_cfg
```

Hay que asegurar que el módulo de drbd se cargue al inicio de sistema en ambos nodos:

```
# echo drbd >/etc/modules-load.d/drbd.conf
```

Ahora se debe montar el recurso encargado montar el disco replicado, en el /opt/zimbra/ con el sistema de fichero “xfs”, configurando el archivo fs_cfg

```
[root@mercurio ~]# pcs cluster cib fs_cfg

[root@mercurio ~]# pcs -f fs_cfg resource create ZimbraFS
Filesystem device="/dev/drbd1" directory="/opt/zimbra"
fstype="xfs"

[root@mercurio ~]# pcs -f fs_cfg constraint colocation add
ZimbraFS with mailDataClone INFINITY with-rsc-role=Master

[root@mercurio ~]# pcs -f fs_cfg constraint order promote
mailDataClone then start ZimbraFS
```

Ahora se le debe indicar al cluster que el disco debe ser montado en el mismo equipo donde se tiene la dirección IP virtual.

```
[root@mercurio drbd.d]# pcs -f fs_cfg constraint colocation add
ClusterIP with ZimbraFS INFINITY
```

Ahora que la configuración del archivo está completa, se procede a cargar los cambios

```
[root@mercurio drbd.d]# pcs cluster cib-push fs_cfg
```

Se puede ver el estado de los recursos y en que servidor se están ejecutando con el comando:

```
[root@mercurio drbd.d]# pcs status
Cluster name: cloffimedicas
Last updated: Sat May 21 18:24:42 2016          Last change: Sat
May 21 18:22:58 2016 by root via cibadmin on mercurio
Stack: corosync
Current DC: mercurio (version 1.1.13-10.el7_2.2-44eb2dd) -
partition with quorum
2 nodes and 4 resources configured

Online: [ hermes mercurio ]

Full list of resources:
```

```
 ClusterIP      (ocf::heartbeat:IPaddr2):       Started mercurio
 Master/Slave Set: mailDataClone [mailData]
     Masters: [ mercurio ]
     Slaves: [ hermes ]

 ZimbraFS       (ocf::heartbeat:Filesystem):    Started mercurio
PCSD Status:
  mercurio: Online
  hermes: Online
```

6 INSTALACIÓN DE ZIMBRA COLLABORATION OS 8.6

6.1 INSTALACIÓN DEL SOFTWARE ZIMBRA

Luego de verificar que los discos de DRBD estén correctamente sincronizados, se puede proceder con la instalación de Zimbra. Para que los servicios se puedan ejecutar en ambos servidores se requiere que la instalación se haga en cada servidor secuencialmente, es decir uno después del otro. Esto con el fin de que los demonios que ejecutan los servicios queden correctamente instalados y no sea solamente una copia de la información.

El proceso de instalación de Zimbra se debe realizar luego de haber instalado correctamente los programas necesarios para su funcionamiento mencionados como pre-requisitos (nc, perl-core, sysstat sqlite, sudo, nc). Este consta de descomprimir un paquete con extensión tgz, el cual me crea una carpeta en el directorio con script llamado "install.sh".

```
[root@mercurio zimbra]# tar xvzf zcs-8.6.0_GA_1153.RHEL7_64.20141215151110.tgz
```

Para iniciar el proceso de instalación se verifica que el disco de DRBD esté montado en "/opt/zimbra" con el comando `df -h`:

```
Filesystem                      Size  Used Avail Use% Mounted on
/dev/drbd1                      3.8T  4.1G  3.8T   0% /opt/zimbra
```

En mercurio se corre el ejecutable install.sh el cual abre una guía de instalación paso a paso de Zimbra y se siguen los siguientes pasos:

- Lo primero por hacer es aceptar el acuerdo de licencia.
  ```
  Do you agree with the terms of the software license
  agreement? [N] y
  ```

- Luego se escogen los paquetes de zimbra que quiere incluir en la instalación, que este caso se deben seleccionar todos.
  ```
  Select the packages to install

  Install zimbra-ldap [Y]
  Install zimbra-logger [Y]
  Install zimbra-mta [Y]
  Install zimbra-dnscache [Y]
  Install zimbra-snmp [Y]
  Install zimbra-store [Y]
  Install zimbra-apache [Y]
  Install zimbra-spell [Y]
  Install zimbra-memcached [Y]
  Install zimbra-proxy [Y]
  ```

- Luego se acepta que se modifique el sistema

```
The system will be modified.  Continue? [N] y
```

- Según como esté configurado las direcciones en el servidor DNS, aparece la opción de cambiar el nombre de dominio. Se hace el cambio de correo.offimedicas.com por offimedicas.com como se observa a continuación:

```
DNS ERROR resolving MX for correo.offimedicas.com
It is suggested that the domain name have an MX record
configured in DNS
Change domain name? [Yes] yes
Create domain: [correo.offimedicas.com] offimedicas.com
        MX: correo.offimedicas.com (10.131.210.50)
de
        Interface: 127.0.0.1
        Interface: ::1
        Interface: 10.131.210.50
done.
```

- En seguida aparece un menú donde se configuran las opciones de red, nombres y contraseñas de las cuentas básicas y la zona horaria entre otros ajustes.

  En la opción de "*Common Configuration*" se cambia la zona horaria, por América/Bogotá y se modifica el nombre del host por correo.offimedicas.com. El nombre del host debe ser igual en ambos servidores y estar configurado en el archivo "hosts" del sistema operativo apuntando a la dirección IP virtual compartida.

  En la opción de "*zimbra store*" se cambian los nombres de las cuentas de spam, ham[7] y la contraseña de la cuenta admin. En zimbra-proxy" se dejan los puertos predeterminados y se cambia el parámetro "*Proxy server mode" a "redirect" para que cuando una petición web se haga sin cifrado, este la direccione al puerto 443 con cifrado.*

Así finaliza la ejecución del script, y ya se tiene instalado el correo en uno de los servidores. Para hacer la instalación en el otro servidor es necesario montar el disco DRBD en /opt/zimbra del nodo secundario del cluster. La manera más sencilla de hacer esto es simular una caída en el servidor donde se hizo la instalación con el siguiente comando:

```
[root@mercurio ~]# pcs cluster standby mercurio
```

[7] Ham es el correo deseado y no es considerado spam, en Zimbra se usa éste término para la cuenta donde se almacena los mensajes que son reconocidos como spam, pero el usuario los desmarca de ésta categoría.

De nuevo se verifica que los recuros se estén ejecutando en Hermes, con "*pcs status*" y que el disco DRBD esté montado en */opt/zimbra/.* Se realiza la instalación en Hermes utilizando exactamente la misma configuración del servidor Mercurio. A pesar de que el disco donde Zimbra guarda la información está replicado, se debe realizar la instalación para que el equipo pueda ejecutar los servicios alojados en el disco, por esto al correr el script de instalación, se preguntará si desea eliminar los archivos instalados en el directorio, y se debe responder afirmativamente.

```
Would you like to delete /opt/zimbra before installing? [N] y
```

Al finalizar la instalación en Hermes, se debe restablecer el estado de Mercurio utilizando el comando:

```
[root@mercurio ~]# pcs cluster unstandby mercurio
```

De esta forma inicia la sincronización de datos entre ambos nodos, teniendo en cuenta que Hermes tiene ahora el rol primario, y por tanto Mercurio adoptará los datos escritos en Hermes.

En caso de que los discos de DRBD no se encuentren conectados, se debe realizar un procedimiento llamado "*Split Brain Recovery"* que consiste en volver a sincronizar los discos luego de que alguno de ellos sufriera inconsistencias en la información, descartando los datos diferentes y copiando lo que haga falta. Se hace de la siguiente manera:

- En ambos nodos

  ```
  drbdadm disconnect zimbradata
  ```

- En Hermes

  ```
  drbdadm secondary zimbradata

  drbdadm connect --discard-my-data zimbradata
  ```

- En Mercurio

  ```
  drbdadm connect zimbradata
  ```

Para verificar el proceso se usa el comando "`cat /proc/drbd`" :

```
[root@mercurio ~]# cat /proc/drbd
version: 8.4.7-1 (api:1/proto:86-101)
GIT-hash: 3a6a769340ef93b1ba2792c6461250790795db49 build by
phil@Build64R7, 2016-01-12 14:29:40

 1: cs:Connected ro:Secondary/Primary ds:UpToDate/UpToDate C r----
-
    ns:0 nr:4008016 dw:4008016 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0
ep:1 wo:f oos:0
```

Para acceder a la consola de administración web, se ingresa con un navegador a la URL https://correo.offimedicas.com:7071 y se digita el usuario “admin” y la contraseña establecida en la instalación.

Si se quiere revisar el estado de la aplicación, por medio de la terminal, desde el usuario zimbra se ejecuta el comando:

```
[zimbra@hermes ~]$ zmcontrol status
Host correo.offimedicas.com
        amavis                  Running
        antispam                Running
        antivirus               Running
        dnscache                Running
        ldap                    Running
        logger                  Running
        mailbox                 Running
        memcached               Running
        mta                     Running
        opendkim                Running
        proxy                   Running
        service webapp          Running
        snmp                    Running
        spell                   Running
        stats                   Running
        zimbra webapp           Running
        zimbraAdmin webapp      Running
        zimlet webapp           Running
        zmconfigd               Running
```

6.2 AÑADIR RECURSO DE MONITOREO DE ZIMBRA

Para instalar el recurso es necesario descargar un script que pueda monitorear el estado de los servicios de Zimbra. El Utilizado por el proyecto fue desarrollado por Adrian Gibanel y compartido a través de GitHub[8]. Este archivo es ubicado en el directorio /usr/lib/ocf/resource.d/btactic/ en ambos nodos y para agregarlo se usan los siguientes pasos:

Crear archivo con configuración actual de Pacemaker.

```
[root@mercurio ~]# pcs cluster cib zimbra_cfg
```

Luego se agrega a éste archivo los parámetros de configuración del recurso. El nombre es ZimbraService y el tiempo de monitoreo de 2 minutos hacen parte de los ajustes.

[8] https://github.com/adrian15/hazimbra-thesis/blob/master/ocf/zimbra

```
[root@mercurio ~]# pcs -f zimbra_cfg resource create ZimbraService
ocf:btactic:zimbra op monitor interval=2min timeout="40s" op start
interval="0s" timeout="360s" op stop interval="0s" timeout="360s"
```

También se especifica que el recurso ZimbraService se ejecute en el nodo donde se está ejecutando el recurso ZimbraFS y que inicie después de éste último.

```
[root@mercurio ~]# pcs -f zimbra_cfg constraint colocation add
ZimbraService with ZimbraFS INFINITY with-rsc-role=Master

[root@mercurio ~]# pcs -f zimbra_cfg constraint order promote
ZimbraFS then start ZimbraService
```

Y se carga el archivo de los parámetros.

```
[root@mercurio ~]# pcs cluster cib-push zimbra_cfg
```

Se verifica el estado del recurso

```
[root@mercurio ~]# pcs status
Cluster name: cloffimedicas
Last updated: Sun May 22 16:42:20 2016          Last change: Sun
May 22 15:36:39 2016 by root via cibadmin on mercurio
Stack: corosync
Current DC: hermes (version 1.1.13-10.el7_2.2-44eb2dd) - partition
with quorum
2 nodes and 5 resources configured

Online: [ hermes mercurio ]

Full list of resources:

 ClusterIP      (ocf::heartbeat:IPaddr2):       Started hermes
 Master/Slave Set: mailDataClone [mailData]
     Masters: [ hermes ]
     Slaves: [ mercurio ]
 ZimbraFS       (ocf::heartbeat:Filesystem):    Started hermes
 ZimbraService  (ocf::btactic:zimbra):          Started hermes

PCSD Status:
  mercurio: Online
  hermes: Online

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

7 IMPLEMENTACIÓN DE REGLAS DE SEGURIDAD PARA EL SERVICIO DE CORREO UTILIZANDO LA CONFIGURACIÓN DE ZIMBRA OS VERSIÓN 8.6

Para aplicar las reglas de seguridad se seccionan las cuentas de correo en diferentes clases de servicio (COS), dependiendo de la necesidad y del rol del usuario. Para los empleados en general se crea una COS llamada “general” (ver Ilustración 11) con el mayor número de restricciones, sin entorpecer el desarrollo de sus funciones, y para los cargos administrativos y gerencia se crea una clase de servicio llamada “admin”, la cual otorga privilegios adicionales. De tal forma que cada vez que se crea una nueva cuenta de correo se debe seleccionar la COS a la que pertenezca.

A continuación, se describe la configuración seleccionada para cada una de las clases de servicio.

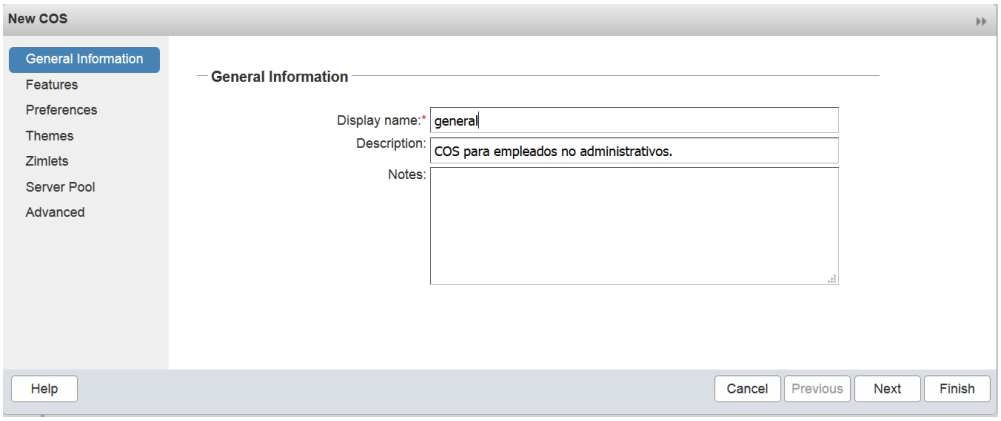

Ilustración 11. Creación de COS general

Dentro de las características de correo electrónico, solamente se deshabilita la opción de acceso externo utilizando el protocolo POP (Ilustración 12. Bloqueo acceso externo POP), con el fin de evitar que los usuarios utilicen algún software de cliente de correo, permitiendo el manejo de información sin registro debido al comportamiento de este protocolo, el cual elimina los datos del servidor una vez el cliente descargue el mensaje.

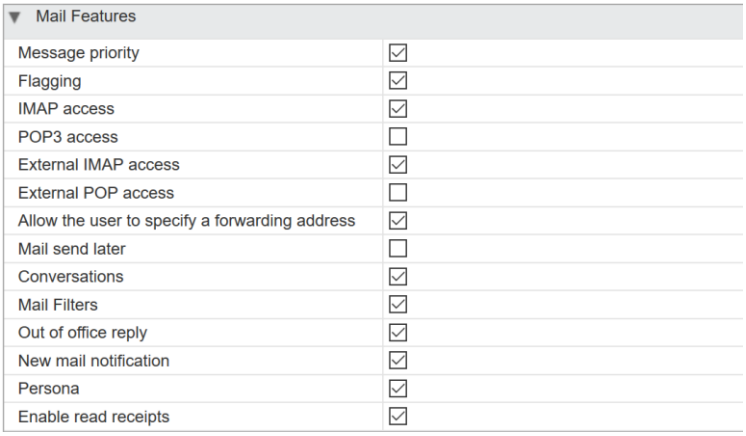

Ilustración 12. Bloqueo acceso externo POP

En los ajustes de contraseña se configuran las características de las claves que definen el nivel de seguridad que se maneja para el acceso a cada cuenta.

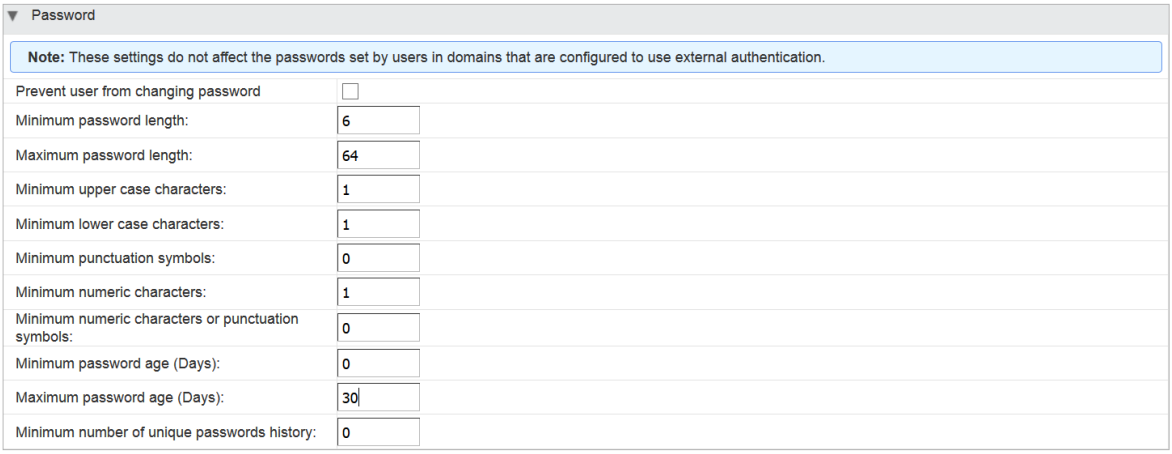

Ilustración 13. Configuración contraseñas para "general"

En la Ilustración 13. Configuración contraseñas para "general", se aprecian los ajustes para la COS “general”, en donde las reglas aplicadas no son tan estrictas. Las condiciones para crear una clave quedaron asignadas de tal forma que los usuarios deben utilizar por lo menos 6 caracteres entre los cuales debe haber al menos un carácter en minúscula, uno en mayúscula y un número, además están obligados a cambiarla cada 30 días como máximo.

Para el personal del área administrativa las condiciones deben ser diferentes, con un nivel de seguridad más alto, por esta razón la extensión mínima pasa a ser de 10 caracteres en una combinación entre letras en minúscula, mayúscula, números y signos de puntuación. La caducidad de la clave será de 30 días.

Para todo el personal se ajustan los parámetros de la política de fallo en el registro, definiendo que el máximo número de intentos fallidos durante una hora para ingresar a la cuenta son 5, antes del bloqueo de la cuenta. Ver Ilustración 14

Ilustración 14. Políticas de registro fallido

El manejo de la caducidad del registro de la sesión se configura con las reglas de *timeout*, donde se establece que cualquier sesión abierta de una cuenta de correo será cerrada después de una hora de inactividad y que las credenciales se mantendrán registradas por un periodo máximo de 4 horas. De igual forma, para la sesión en la interfaz de administración el tiempo máximo de registro es de 4 horas. De este modo se restringe el acceso de intruso a cuentas ajenas, en el caso de dejar la sesión abierta después de terminar una jornada laboral. Ver (Ilustración 15)

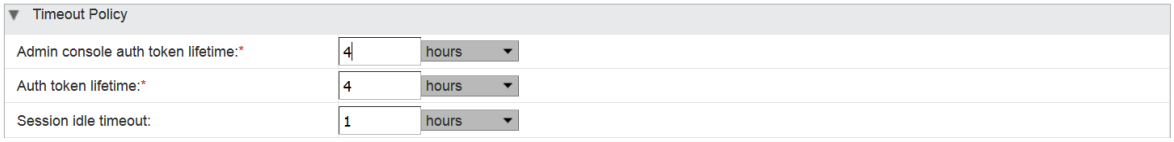

Ilustración 15. Política de Timeout

Dentro de los archivos adjuntos se envían gran cantidad de virus y la versión de Zimbra que se está utilizando ofrece la opción de bloquear todos los adjuntos, pero esto no es una idea que se pueda aplicar en Offimedicas. En cambio, existe un parámetro de configuración donde se pueden bloquear extensiones de archivos adjuntos partiendo de una lista por defecto de 27 tipos que se pueden agregar las que se deseen. En el proyecto inicialmente se bloquean todos los tipos sugeridos como se muestra en la Ilustración 16.

Cuando se intenta enviar o recibir un mensaje con archivos adjuntos con alguna de las extensiones bloqueadas, el usuario de Zimbra será notificado a través de un mensaje de aviso que no se pudo completar el envío.

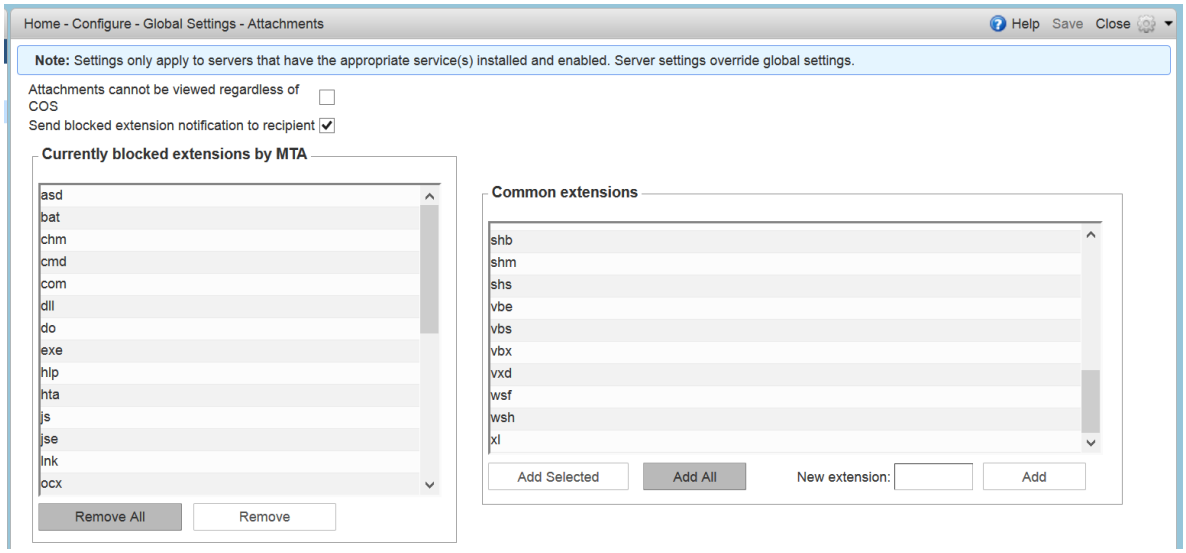

Ilustración 16. Extensiones de archivos adjuntos bloqueadas

Dentro de los parámetros de configuración global existe una sección denominada MTA, en donde se ajustan valores de autenticación, conexión de red y políticas de confirmación entre otras. Para el desarrollo del proyecto se escogió la siguiente configuración:

Se habilitan las opciones de “Activar autenticación” que le permite a software de clientes de correo comunicarse con el MTA de Zimbra. La opción “Sólo autenticación TLS” obliga a usar TLS (*Transaction Level Security*), similar a SSL, para evitar el envío de claves sin cifrar. (Zimbra, Inc, 2014)

Por defecto, la versión utilizada de Zimbra, está configurada para no aceptar ningún tipo de Relay, por lo que estos valores no se modifican.

Ilustración 17. Chequeo de protocolos

En la Ilustración 17, se muestra la selección de las 3 opciones para la revisión protocolos, donde la primera se refiere al rechazo de la solicitud del cliente cuando maneja una sintaxis inválida para el estándar de comunicación. El segundo parámetro está relacionado con el rechazo de la solicitud de conexión cuando el cliente no salude con un FQDN, y el último ítem define el rechazo de las solicitudes cuando la dirección del remitente no contiene un FQDN.

La revisión de DNS es una herramienta más de Zimbra para comprobar la confiabilidad de los servidores remitentes de mensajes. En la Ilustración 18, se visualiza seleccionada la primera opción (*Client´s IP address*) con la cual se verifica que no exista ningún error en el mapeo de dirección-nombre, o nombre-dirección del remitente. En el caso de encontrar una anomalía se rechaza la solicitud del cliente. Con el parámetro *Hostname in greeting*, se rechaza la solicitud si es cliente no tiene registros A o MX. Por último, con la tercera opción refuta la solicitud cuando el servidor no es el destino final para el remitente.

Ilustración 18. Revisión de DNS

Los ajustes de Anti-spam se dejan con los valores por defecto (ver Ilustración 19) que trae la versión implementada de Zimbra, los cuales se basan en las reglas predefinidas por la base de datos Bayes[9] para calificar cada uno de los mensajes recibidos. Zimbra evalúa los correos basándose en porcentajes, si un mensaje está etiquetado entre un 33% y un 75% es considerado spam y depositado en la carpeta de no deseados. Mientras que los mensajes que estén por encima del 75% son descartados antes de enviarlos al cliente destino.

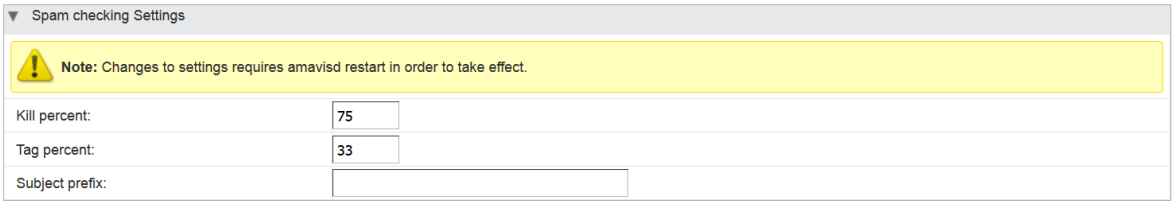

Ilustración 19. Configuración Anti-spam

[9] https://spamassassin.apache.org/full/3.0.x/dist/doc/sa-learn.html

8 PRUEBAS DE FUNCIONAMIENTO

Antes de poner en funcionamiento el cluster de alta disponibilidad para el correo, es necesario asegurar que todo esté funcionando correctamente. Para realizar las pruebas de seguridad se utiliza un dominio diferente al de offimedicas.com. Esto es necesario debido a que no es posible recibir mensajes externos en el servidor configurado en el proyecto, mientras el servidor actual de Offimedicas está en funcionamiento, por lo tanto, se implementa el dominio selimsaba.com para la realización de estas pruebas.

8.1 PRUEBAS DE SERVICIO

Acceder al servicio de correo es posible mediante un navegador web, ingresando la dirección URL correo.offimedicas.com, donde se direccionará a https://correo.offimedicas.com y abrirá un formulario para digitar el usuario y la contraseña de las cuentas de correo, como se muestra en la Ilustración 20.

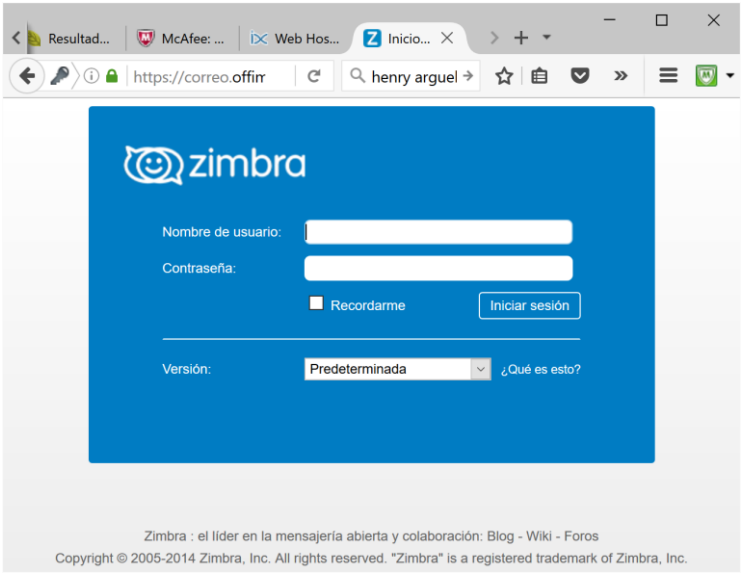

Ilustración 20. Acceso web

Si se requiere crear cuentas o modificar ajustes de configuración, se pueden realizar por la consola de administración, accediendo a https://correo.offimedicas.com:7071. Con la Ilustración 21 se evidencia la interfaz de la consola de administración.

Por el cliente web también es posible acceder a la “Agenda”, “Maletín”, “Contactos”, “Tareas” y “Preferencias”, cambiando las pestañas que están abajo del logo de Zimbra. En la Ilustración 22 se observa la interfaz web con la pestaña de correo seleccionada y las otras opciones al lado derecho de esta.

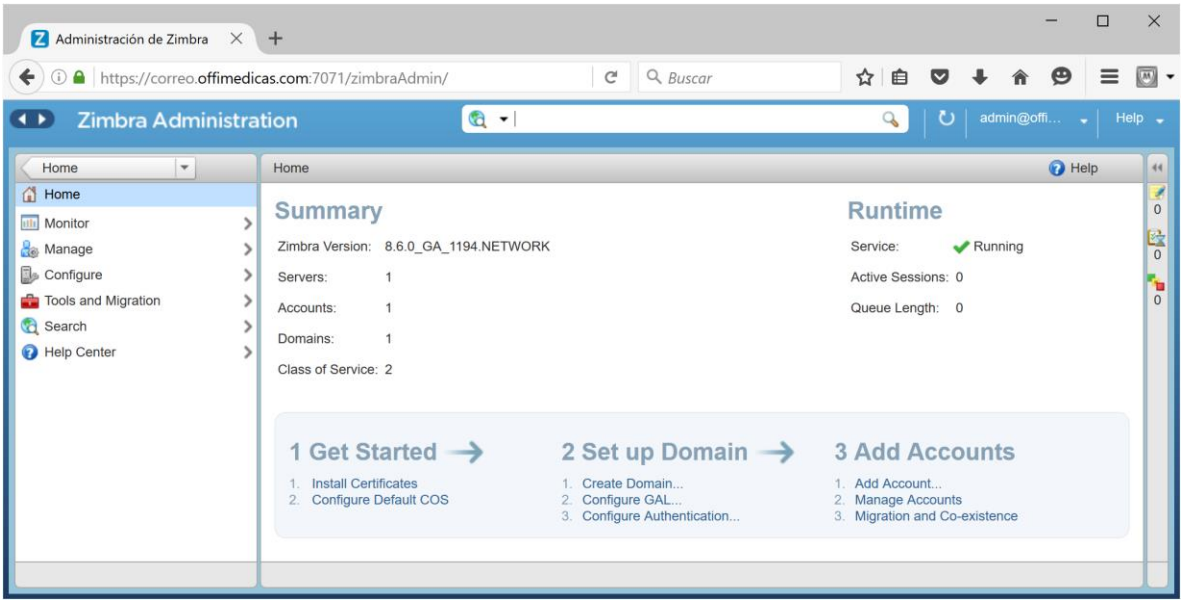

Ilustración 21. Consola de administración

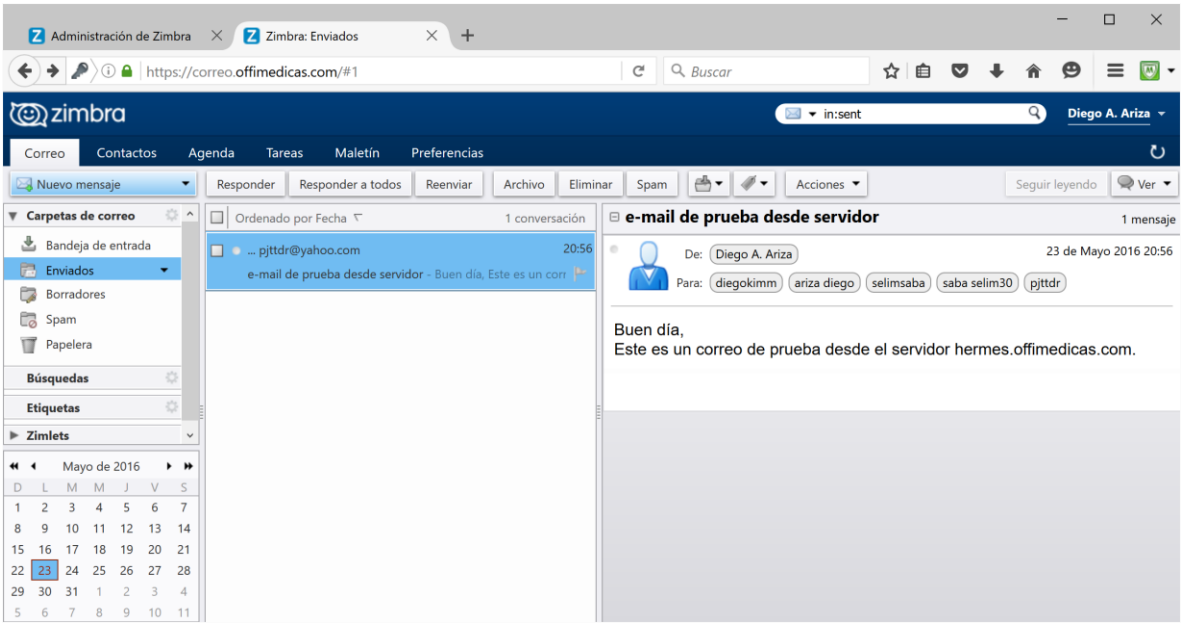

Ilustración 22. Interfaz de correo

8.2 PRUEBAS DE SEGURIDAD

Las pruebas de seguridad toman en consideración la capacidad de un intruso de penetrar la red y la configuración de los registros de autenticación en el servidor DNS. Para las pruebas de penetración y configuración se usan páginas web especializadas en este asunto.

En el sitio web www.mail-tester.com, se puede enviar un mensaje de correo electrónico y este analiza la configuración en lo servidores DNS, revisa el puntaje de SpamAssassin, revisa si el dominio o la dirección IP está reportado en algún RBL y con base en esto calcula un puntaje y estima la posibilidad de que el destinatario reciba el correo en la bandeja de entrada.

Se hizo la prueba enviando un mensaje al correo web-uiFTbw@mail-tester.com y se obtuvo un puntaje 9/10 (ver Ilustración 23), indicando que con probablemente los mensajes enviados desde offimedicas.com sean recibidos en la bandeja de entrada y no en la carpeta de correo no deseado. Los resultados de la prueba están disponibles en “https://www.mail-tester.com/web-uiFTbw”.

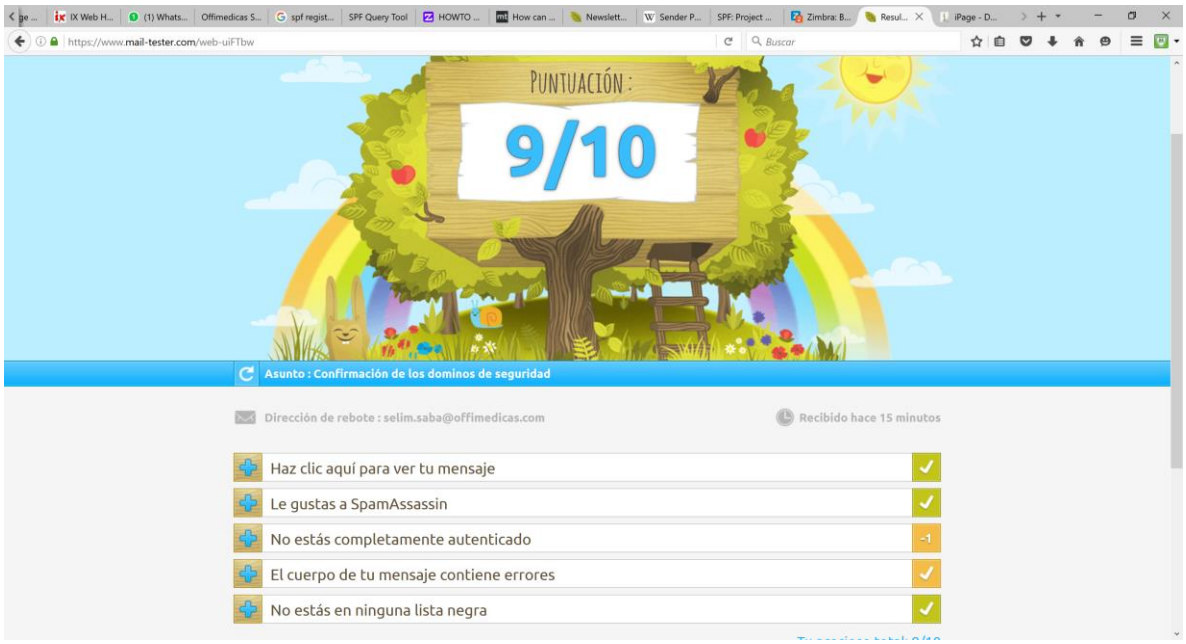

Ilustración 23. Prueba de configuración.

Los resultados de esta prueba muestran que hace falta configurar los registros DMARC, en los registros TXT de los servidores de dominio, sin embargo, también se necesario realizar esta configuración en Zimbra. Mediante el comando “`/opt/zimbra/libexec/zmdkimkeyutil -a -d offimedicas.com`”, se puede configurar las firmas DKIM, lo que permite autenticar que el correo proviene de offimedicas.com.

Aunque Zimbra tenga esta opción, es necesario poder agregar la firma que genera en el servidor DNS, y el proveedor ixwebhosting.com no permite hacerlo. Por esto, no se firman los correos con DKIM.

En las pruebas de penetración, se usó el sitio web http://www.emailsecuritycheck.net el cual hace una prueba enviando siete mensajes de correo electrónico a una dirección que le sea provista. Estos siete mensajes contienen archivos y mensajes que el servidor debería detectar como maliciosos y no entregarlos al destinatario.

En la prueba se usó como e-mail de ejemplo selim.saba@selimsaba.com, al cual llegaron los mensajes de contenido prohibido y filtro de correo desde el servido Mercurio, como se ve en la Ilustración *24*. De los siete mensajes enviados cinco tenían extensiones prohibidas, uno contenía un archivo comprimido identificado como virus y otro tenía una firma que es reconocida como SPAM. El mensaje marcado como SPAM no se recibió en la bandeja de entrada, pero tampoco de se recibió en la bandeja de correo no deseado por haber superado el porcentaje de SpamAssassin configurado para estas.

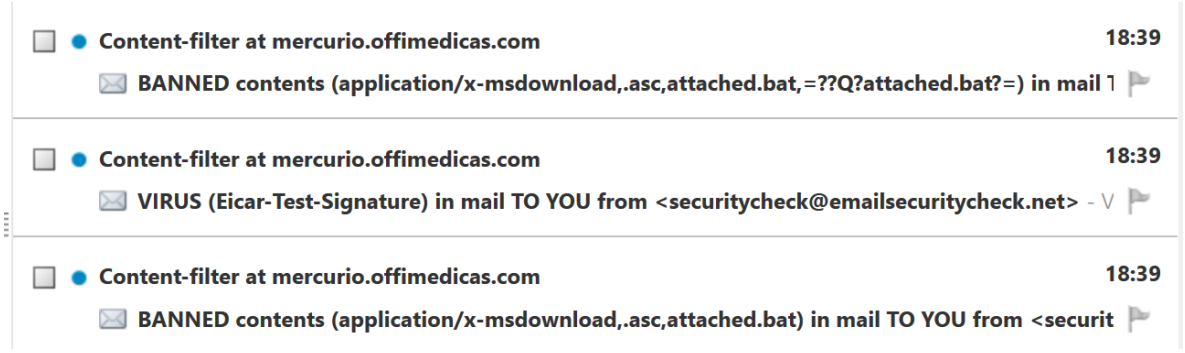

Ilustración 24. Contenido filtrado

En el sitio “http://emailsecuritygrader.com/”, se pueden realizar pruebas que muestran si el servidor puede ser usado como relevo de correo (Mail Relay), obteniendo un resultado exitoso al no permitir el este mecanismo. (Ver Ilustración 25)

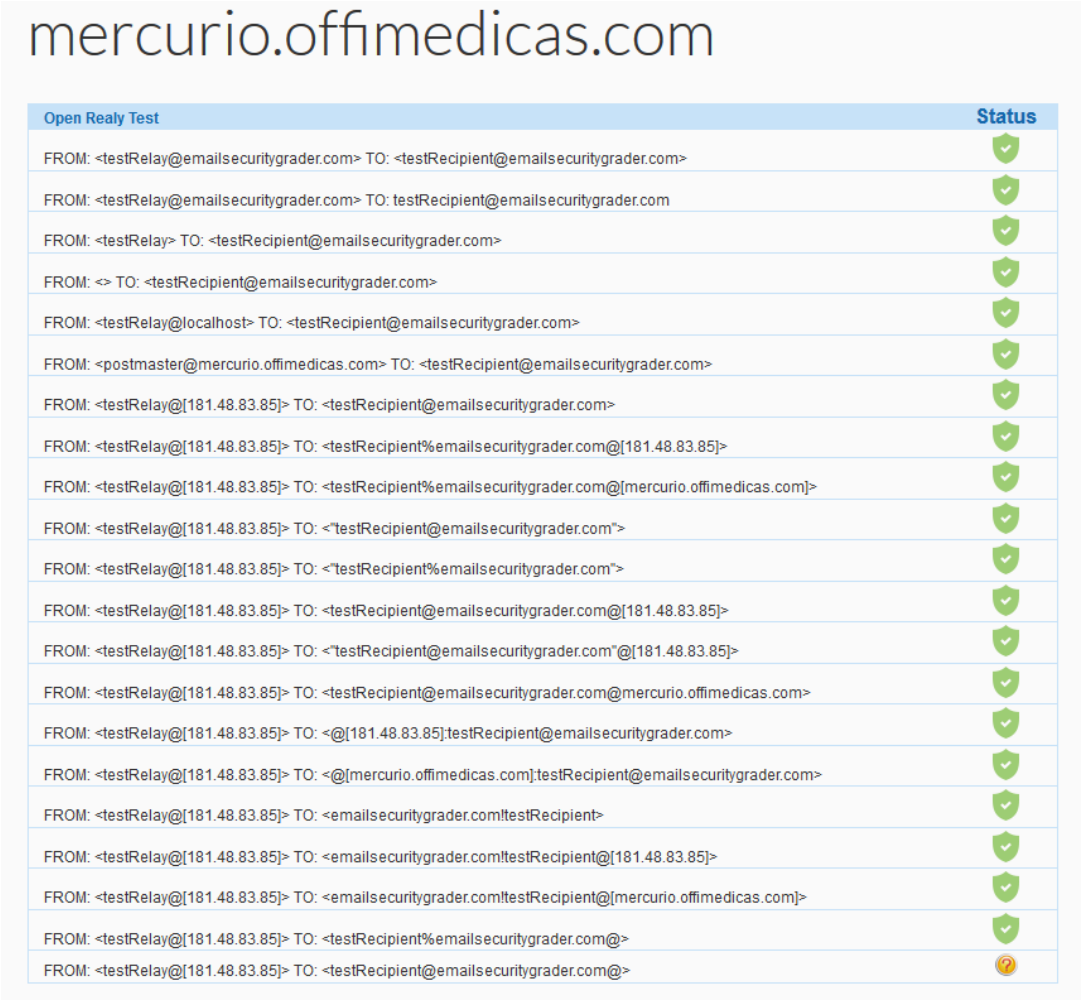

Ilustración 25. Prueba de Mail Relay.

8.3 LAS PRUEBAS DE ALTA DISPONIBILIDAD

La alta disponibilidad se puede probar desconectando el servidor que aloja los servicios, así, los recursos deben montarse automáticamente en el otro servidor. En Mercurio se corre la IP virtual y el disco /opt/zimbra como se evidencia a continuación:

```
[root@mercurio ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP qlen 1000
    link/ether 08:00:27:0d:b2:e7 brd ff:ff:ff:ff:ff:ff
```

```
    inet 10.131.210.51/24 brd 10.131.210.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet 10.131.210.50/24 brd 10.131.210.255 scope global
secondary enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe0d:b2e7/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP qlen 1000
    link/ether 08:00:27:08:70:fa brd ff:ff:ff:ff:ff:ff
    inet 192.168.99.1/30 brd 192.168.99.3 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe08:70fa/64 scope link
       valid_lft forever preferred_lft forever

[root@mercurio ~]# df -h
Filesystem                          Size  Used Avail Use% Mounted on
/dev/mapper/centos_mercurio-root  2.0G  1.4G  674M  67% /
devtmpfs                            1.9G     0  1.9G   0% /dev
tmpfs                               1.9G   39M  1.9G   3% /dev/shm
tmpfs                               1.9G  8.4M  1.9G   1% /run
tmpfs                               1.9G     0  1.9G   0%
/sys/fs/cgroup
/dev/sda1                           497M  163M  334M  33% /boot
tmpfs                               380M     0  380M   0%
/run/user/0
/dev/drbd1                          4.1T  3.9G  4.1T   1% /opt/zimbra
```

Si se apaga Mercurio entonces en Hermes deben aparecer estas mismas características:

[root@hermes ~]# ip a

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP qlen 1000
    link/ether 08:00:27:84:64:1a brd ff:ff:ff:ff:ff:ff
    inet 10.131.210.52/24 brd 10.131.210.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet 10.131.210.50/24 brd 10.131.210.255 scope global
secondary enp0s3
```

```
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe84:641a/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP qlen 1000
    link/ether 08:00:27:85:72:ec brd ff:ff:ff:ff:ff:ff
    inet 192.168.99.2/30 brd 192.168.99.3 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe85:72ec/64 scope link
       valid_lft forever preferred_lft forever

[root@hermes ~]# df -h
Filesystem                      Size  Used Avail Use% Mounted on
/dev/mapper/centos_hermes-root  2.0G  1.4G  669M  68% /
devtmpfs                        1.9G     0  1.9G   0% /dev
tmpfs                           1.9G   54M  1.8G   3% /dev/shm
tmpfs                           1.9G  8.4M  1.9G   1% /run
tmpfs                           1.9G     0  1.9G   0%
/sys/fs/cgroup
/dev/sda1                       497M  163M  334M  33% /boot
tmpfs                           380M     0  380M   0% /run/user/0
/dev/drbd1                      4.1T  3.9G  4.1T   0% /opt/zimbra
```

9 CONCLUSIONES

Cambiando la infraestructura del correo electrónico, que está puesta en marcha sobre un equipo único, a un cluster de alta disponibilidad con dos nodos, se elimina el punto crítico de fallo, mejorando la probabilidad de mantener el servicio en funcionamiento. Cuando se elimina el punto crítico de fallo se requiere que exista más de un elemento sin actividad para que se interrumpa el servicio, dando la posibilidad de tomar acciones correctivas sobre el punto afectado sin que el usuario lo note.

Los registros MX, SPF y A, deben estar debidamente configurador en los servidores DNS públicos, para que los mensajes enviados a otros dominios, puedan ser identificados y no se descarten como mensajes de SPAM. Sin configurar estos, algún falsificador podría suplantar la identidad. Zimbra cuenta con un mecanismo de autenticación llamado "DomainKeys Identified Mail" que permite firmar los mensajes usando una llave pública alojada en los servidores DNS, sin embargo, no todos los proveedores de dominios de internet, aceptan la incorporación de éste tipo de registro en sus servidores de nombres dominio.

La versión gratuita de Zimbra, no tiene integrada la opción de implementar alta disponibilidad, por lo que requiere de software adicional para lograr este objetivo. DRBD y Pacemaker son opciones de código abierto y robustas con las que se puede lograr este objetivo. Una ventaja de estas herramientas es el soporte brindado por las páginas oficiales y la comunidad de usuarios, esta información tiene gran valor gracias a su constante actualización.

Para el montaje de un cluster de alta disponibilidad se debe contar con al menos dos equipos de cómputo que tengan características semejantes en hardware e idénticas en software, de tal forma que cualquiera de los nodos pueda cumplir el papel de servidor sin que el usuario final note alguna diferencia en la prestación del servicio.

No es posible eliminar todas las fuentes de riesgo para los servidores y clientes de correo, ya que el uso de algunas funciones primordiales de este tipo de servicio, como el envío de archivos adjuntos, abre una brecha de seguridad. Por otro lado, la eficacia del antivirus y del filtro anti-spam no es del cien por cien, lo cual deja en manos del usuario la tarea de identificar estas amenazas y eliminarlas.

El software de correo Zimbra Collaboration OS ofrece un conjunto de herramientas, adicionales al envío y recepción de mensajes por correo electrónico. Dichas herramientas permiten eliminar el uso de software adicional que cumplen funciones semejantes, por ejemplo, el almacenamiento de archivos en la nube se puede hacer a través del "Maletín", y el seguimiento de actividades de proyectos se podría manejar por medio de las "Tareas".

Este trabajo de aplicación puede ser implementado en cualquier empresa que desee manejar su propio correo electrónico o prestar éste servicio a terceros, porque cuenta con alta disponibilidad, seguridad en la información y autonomía en la administración a un bajo costo.

BIBLIOGRAFÍA

Alta disponibilidad. (s.f.). (ccm) Recuperado el 01 de 03 de 2016, de http://es.ccm.net/contents/634-alta-disponibilidad

Apache Foundation. (2016). *SpamAssassin Wiki.* Recuperado el 10 de 04 de 2016, de http://wiki.apache.org/spamassassin/SpamAssassin

Bhardwaj, J. (07 de Nov de 2011). *Naked security*. (sophos.com) Recuperado el 01 de Marzo de 2016, de https://nakedsecurity.sophos.com/es/2012/11/07/email-systems-are-fundamentally-insecure/

Crispin, M. (Marzo de 2003). *RFC 3501 - INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1*. (IETF) Recuperado el 01 de 03 de 2016, de https://tools.ietf.org/html/rfc3501

Freed, N., & Borenstein, N. (Noviembre de 1996). *RFC 2045 - Multipurpose Internet Mail Extensions*. (IETF) Recuperado el 01 de Marzo de 2016, de https://tools.ietf.org/html/rfc2045

Fuentes Serrano, L. F., & Calderón Hernández, C. (s.f.). *Phishing Scam*. (UNAM) Recuperado el 01 de Marzo de 2016, de http://www.seguridad.unam.mx/usuario-casero/eduteca/main.dsc?id=166

Hellman, B., Haas, F., Reisner, P., & Ellenberg, L. (2011). *DR:DB.* Recuperado el Abril de 2016, de https://www.drbd.org/en/

How PGP works. (s.f.). (PGPI) Recuperado el 01 de Marzo de 2016, de http://www.pgpi.org/doc/pgpintro/

Klensin, J. (Octubre de 2008). *RFC 5321 - Simple Mail Transfer Protocol*. (IETF) Recuperado el 01 de 03 de 2016, de https://tools.ietf.org/html/rfc5321

Myers, J., Mellon, C., & Rose, M. (Mayo de 1996). *Post Office Protocol - Version 3*. (IETF) Recuperado el 01 de 03 de 2016, de https://tools.ietf.org/html/rfc1939

Securing SMTP mail servers. (29 de Marzo de 2010). (techworld) Recuperado el 01 de Marzo de 2016, de http://www.techworld.com/tutorial/security/securing-smtp-mail-servers-408/

Spam. (s.f.). Recuperado el 01 de Marzo de 2016, de http://www.seguridadpc.net/spam.htm

Zimbra, Inc. (Diciembre de 2014). *zimbra.* Recuperado el 15 de Mayo de 2016, de https://files.zimbra.com/website/docs/8.6/Zimbra_OS_Admin_Guide_8.6.0.pdf

Zimbra, Inc. (Diciembre de 2014). *zimbra.* Recuperado el Abril de 2016, de https://www.zimbra.com/documentation/