

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR



FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS

**DISERTACIÓN DE GRADO PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**

**GUÍA METODOLÓGICA PARA LA IMPLEMENTACIÓN DE
UN CLÚSTER DE ALTA DISPONIBILIDAD PARA
SERVIDORES DE APLICACIONES, UTILIZANDO OPEN
SOURCE. CASO DE ESTUDIO: EMPRESA SOPORTE LIBRE**

Gabriel Andrade Pazmiño

2015, MAYO

QUITO - ECUADOR

DEDICATORIA

Este trabajo se lo dedico a mi ñaña Cami, mi angelito que siempre está presente. Ella es mi principal motivación para seguir adelante, que siempre desde el cielo me protege y me ilumina para tomar las mejores decisiones que me permiten cumplir con todos los objetivos que me propongo y así ser una mejor persona en todos los aspectos de mi vida. También quisiera dedicar este trabajo a mis papás, porque han sido un ejemplo a seguir, que al ver su esfuerzo, dedicación y ganas en todo lo que hacen, me ha hecho querer aplicar todas esas virtudes en este trabajo y en todas las acciones de la vida.

AGRADECIMIENTOS

En primer lugar quiero dar un especial agradecimiento a Soporte Libre Cia. Ltda., que durante este proceso ha sido un gran fundamento, porque no solo me permitió aplicar mis conocimientos en sus procesos, pero más que todo fueron una gran fuente de enseñanza, lo cual ha contribuido a mi formación como profesional. En especial quisiera agradecer a Jorge Armijo, que durante el desarrollo de esta disertación de grado fue un gran soporte, brindándome todo su apoyo técnico y también su amistad. Así mismo quisiera agradecer a mi directora Beatriz Campos, a mis correctores, la ing. Suyana Arcos y el ing. Alfredo Calderón, por dar su tiempo en la dirección y corrección de esta disertación de grado y por ser una grandes maestros.

Un especial agradecimiento a mis papás por darme la oportunidad de estudiar esta carrera que tanto me gusta, por siempre estar ahí, por su paciencia, y en especial quisiera agradecerles por toda la confianza y amor que me dan.

También me gustaría agradecer a mi prima Cris, por apoyarme en este camino y a toda mi familia, tías, tíos, primas, primos y abuelita que constantemente me han apoyado en la finalización de este trabajo, pero que también con su cariño son un gran soporte diario.

Me gustaría agradecer a mi novia Sofi, que gracias a su cariño y comprensión he tenido todo el empuje y ganas para poder cumplir mis metas.

MUCHAS GRACIAS A TODOS.

RESUMEN

Hoy en día, es vital asegurar el funcionamiento continuo de las aplicaciones, una caída del servicio puede significar grandes pérdidas económicas para los usuarios, además de una pérdida de confianza en la empresa proveedora del servicio. Para evitar que esto suceda se propone la utilización de la metodología para la implementación de un clúster en alta disponibilidad, el cual evita una pérdida de información, manteniendo la integridad en los datos con una mayor velocidad de respuesta, satisfaciendo así las necesidades de los clientes. La Guía Metodológica para la Implementación de un Clúster en Alta Disponibilidad para Servidores de Aplicaciones utilizando Open Source que se presenta en esta disertación de grado, la cual fue creada en base a los conceptos y fundamentos teóricos, permite tener una ruta fija y estándar para la implementación de un clúster en alta disponibilidad. Esta guía metodológica fue diseñada y validada para el uso de Soporte Libre Cía. Ltda., empresa que se verá beneficiada por su utilidad. Los métodos propuestos en esta guía metodológica tuvieron resultados positivos, ya que mediante la validación se comprobó de manera práctica que al seguir los pasos se obtiene una alta disponibilidad en los servidores de aplicaciones Open Source.

La guía realizada es una herramienta facilitadora la cual simplifica el proceso de implementación de servidores de aplicaciones en alta disponibilidad y podría ser utilizada por varias empresas, siempre y cuando tengan todas las herramientas necesarias.

INTRODUCCIÓN

En un mundo competitivo las empresas no pueden tener las aplicaciones sin funcionar, porque esto puede resultar en una pérdida y corrupción de la información que afecta la productividad de la empresa, posiblemente causando pérdidas económicas por falla del sistema. Es necesario que empresas que ofrecen el servicio de Servidores de Aplicaciones tomen en cuenta la importancia de mantener el alto funcionamiento de las aplicaciones para asegurar la calidad de la información.

Una solución actual para contrarrestar la problemática de la no actividad en una aplicación es implementar Alta Disponibilidad en Servidores de Aplicaciones utilizando Open Source.

En este contexto esta disertación de grado propone una herramienta Open Source para que Soporte Libre Cía. Ltda., pueda implementar en sus operaciones y así pueda ofrecer alta disponibilidad en los servicios con las que operan y ofrece a sus clientes.

El objetivo es realizar una guía metodológica la cual indica la ruta y los pasos a seguir para poder implementar un clúster Activo-Activo.

Para el correcto funcionamiento de esta guía es de suma importancia tener un conocimiento profundo de los fundamentos técnicos de los conceptos base del clúster. En el Marco Teórico se los estudiará a profundidad, para posteriormente en el capítulo 2 explicar el ámbito de aplicación de la herramienta en una empresa real de la ciudad de Quito para finalmente en el capítulo 3 y 4 presentar la guía metodológica y su respectiva implementación y validación.

TABLA DE CONTENIDOS.

INTRODUCCION.....	V
CAPITULO 1	4
1. MARCO TEORICO.....	4
1.1. FUNDAMENTOS GENERALES DE CLÚSTER.....	4
1.1.1. CONCEPTO	4
1.1.2. CLASIFICACIÓN.....	5
1.1.2.1. ALTO RENDIMIENTO (HIGH PERFORMANCE).....	5
1.1.2.2. ALTA EFICIENCIA (HIGH THROUGHPUT).....	6
1.1.2.3. ALTA DISPONIBILIDAD (HIGH AVAILABILITY)	6
1.1.3. TIPOS DE IMPLEMENTACIÓN DE CLÚSTERES	6
1.1.3.1. CLÚSTERES ACTIVO-ACTIVO	6
1.1.3.2. CLÚSTERES ACTIVO-PASIVO.....	7
1.1.4. COMPONENTES DE UN CLÚSTER.....	9
1.1.4.1. NODOS	9
1.1.4.2. SISTEMA OPERATIVO.....	10
1.1.4.3. RED DE INTERCONEXIONES DE ALTAS PRESTACIONES.....	10
1.1.4.4. MIDDLEWARE.....	10
1.2. SERVIDOR DE APLICACIONES.....	12
1.2.1. SERVIDOR DE APLICACIONES J2EE	13
1.2.1.1. JBOSS EAP (ENTERPRISE APPLICATION PLATFORM) 6.3	15
1.2.1.2. GLASSFISH.....	15
1.2.2. OTROS TIPOS DE SERVIDORES DE APLICACIONES	16
1.3. BALANCEADOR DE CARGAS	16
1.3.1. BALANCEO DE CARGA EN HARDWARE	16
1.3.2. BALANCEO DE CARGA EN SOFTWARE.....	17
1.3.2.1. ROUND ROBIN	17
1.3.2.2. MÓDULO MOD_CLUSTER.....	17
1.3.2.3. LINUX VIRTUAL SERVER (LVS)	18
CAPITULO 2	19
2. CASO DE ESTUDIO SOPORTE LIBRE.....	19
2.1. HISTORIA.....	19
2.2. MAPA DE PROCESOS	21
2.3. NECESIDAD DE ALTA DISPONIBILIDAD.....	23

CAPITULO 3	25
3. GUIA METODOLOGICA PARA LA IMPLEMENTACION DE UN CLUSTER EN ALTA DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES	25
3.1. CONFIGURACIÓN DE SERVIDOR	26
3.1.1. REQUISITOS MÍNIMOS DE SERVIDOR PARA SOPORTAR RHEL (RED HAT ENTERPRISE LINUX).....	27
3.1.2. CONFIGURACIÓN PLATAFORMA VIRTUALIZACIÓN K.V.M.....	28
3.1.2.1. SELECCIÓN DE MEDIO DE INSTALACIÓN.....	30
3.1.2.2. ASIGNACIÓN DE MEMORIA.....	31
3.1.2.3. ASIGNACIÓN DE ESPACIO DE ALMACENAMIENTO.....	31
3.1.2.4. CONFIGURACIÓN INTERFACES DE RED Y NOMBRE.....	32
3.1.3. INSTALACIÓN DE SISTEMA OPERATIVO RHEL (RED HAT ENTERPRISE LINUX) 33	
3.1.3.1. MENÚ DE ARRANQUE.....	33
3.1.3.2. SELECCIÓN DE IDIOMA Y TECLADO	36
3.1.3.3. DISPOSITIVOS DE ALMACENAMIENTO	37
3.1.3.4. CONFIGURACIÓN DE NOMBRE DE HOST	38
3.1.3.5. CONFIGURACIÓN DE HUSO HORARIO	39
3.1.3.6. ESTABLECIMIENTO DE CONTRASEÑA DE USUARIO ROOT	40
3.1.3.7. CONFIGURACIÓN DEL PARTICIONAMIENTO DEL DISCO	40
3.1.3.8. CONFIGURACIÓN DEL GESTOR DE ARRANQUE.....	43
3.1.3.9. SELECCIÓN DE TIPO DE SERVIDOR	44
3.2. CONFIGURACIÓN DE RED HAT ENTERPRISE LINUX 6.6.....	50
3.2.1. CONFIGURACIÓN DE INTERFAZ DE RED.....	50
3.2.2. CONFIGURACIÓN DE RESOLUCIÓN DE NOMBRES	51
3.2.3. INSTALACIÓN DE PAQUETES.....	51
3.2.3.1. INSTALACIÓN DE REPOSITORIO PROPIO DE DATOS.....	51
3.2.3.2. REGISTRAR SERVIDOR A RED HAT NETWORK.....	54
3.3. CONFIGURACIÓN DE SERVIDOR DE BALANCEO	59
3.3.1. INSTALACIÓN DE PAQUETES NECESARIOS PARA LA CONFIGURACIÓN DEL SERVIDOR DE BALANCEO	59
3.3.2. CONFIGURACIÓN DEL SERVIDOR DE BALANCEO DE CARGA.....	60
3.3.2.1. HTTPD.CONF	60
3.3.2.2. MOD_CLUSTER.CONF	60
3.3.3. CONFIGURACIÓN DE IPTABLES	63
3.3.4. INICIALIZACIÓN DEL APACHE.....	63
3.4. INSTALACIÓN DE JBOSS EAP 6.3.....	64
3.4.1. REQUISITOS PARA FUNCIONAMIENTO DE SERVIDOR DE APLICACIONES	64

3.4.1.1. INSTALACIÓN DE OPENJDK.....	64
3.4.2. INSTALACIÓN DE SERVIDOR DE APLICACIONES JBOSS EAP 6.3	65
3.4.2.1. UTILIZACIÓN DE ARCHIVO COMPRIMIDO (ZIP) DE JBOSS EAP 6.3	65
3.5. ESTRUCTURA DE JBOSS EAP 6.3	66
3.6. UTILIZACIÓN DE JBOSS EAP 6.3 COMO SERVICIO.....	67
3.7. ARRANQUE DE JBOSS EAP 6.3	68
3.7.1. CONFIGURACIÓN DE IPTABLES	69
3.7.2. CREACIÓN DE USUARIO INICIAL PARA CONSOLA DE ADMINISTRACIÓN	69
3.7.3. INICIALIZACIÓN DE JBOSS EAP 6.3	72
3.7.3.1. MECANISMO DE ARRANQUE EN STANDALONE MODE.....	72
3.8. PRUEBA DE FUNCIONAMIENTO JBOSS EAP 6.3	72
3.9. CONFIGURACIÓN DE CLÚSTER EN STANDALONE MODE.....	73
3.10. BENEFICIOS DE LA IMPLEMENTACIÓN DE LA GUÍA METODOLÓGICA PARA CONFIGURAR UN CLÚSTER EN ALTA DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES UTILIZANDO OPEN SOURCE	77
3.11. DIAGRAMA DE FLUJO DE FUNCIONAMIENTO DE CLÚSTER EN ALTA DISPONIBILIDAD JBOSS EAP 6.3	80
CAPITULO 4	82
4. VALIDACION DE LA GUIA METODOLOGICA PARA LA IMPLEMENTACION DE UN CLUSTER EN ALTA DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES.....	82
4.1. IMPLEMENTACIÓN DE MAQUINAS VIRTUALES.....	82
4.2. INSTALACIÓN DE SISTEMA OPERATIVO RED HAT ENTERPRISE LINUX 6.6	82
4.3. CONFIGURACIÓN DE NOMBRE DEL SERVIDOR Y RESOLUCIÓN DE NOMBRES	83
4.4. CONFIGURACIÓN REPOSITORIOS	84
4.5. CONFIGURACIÓN BALANCEADOR	86
4.6. INSTALACIÓN JBOSS EAP 6.3	87
4.7. INSTALACIÓN STANDALONE.....	87
4.8. DESPLIEGUE DE APLICACIÓN	90
4.9. LEVANTAMIENTO DE BALANCEADOR.....	90
4.10. LEVANTAMIENTO DE APLICACIÓN DE PRUEBA	91
4.11. PRUEBAS DE FUNCIONAMIENTO DE ARQUITECTURA CLUSTER EN ALTA DISPONIBILIDAD	94
4.11.1. PRUEBA 1.....	94
4.11.2. PRUEBA 2.....	95
4.11.3. PRUEBA 3.....	98
CONCLUSIONES.....	103
RECOMENDACIONES.....	105

CAPITULO 1

1. MARCO TEORICO.

1.1. Fundamentos Generales de Clúster

1.1.1. Concepto

Un clúster es un sistema paralelo distribuido, el cual consiste en un conjunto de servidores independientes configurados con características comunes que se encuentran interconectados entre si y que trabajan en conjunto como un simple recurso integrado los servidores que son miembros del clúster son también conocidos como “nodos”. (Baker & Buyya, 1999)

Los servidores pueden estar en el mismo centro de datos o estar físicamente separadas pero conectados mediante una red LAN. El clúster de servidores funciona como un sistema único tolerante a fallas o también llamado sistema monolítico, lo cual significa que para el usuario final que consume el servicio que brinda el clúster todos los nodos son vistos como un solo servidor, lo que permite tener una alta disponibilidad de los servicios que se esté conformando.

Los clústeres pueden clasificarse de acuerdo a la configuración de que tengan los nodos, es decir puede haber clúster homogéneos y heterogéneos. En los homogéneos las arquitecturas son parecidas y en todos los nodos corre el mismo sistema operativo, la misma versión de los aplicativos, los mismos servicios en cada uno de los nodos. En los clúster heterogéneos los nodos pueden tener arquitecturas distintas y pueden trabajar con sistemas operativos diferentes.

Un beneficio de la utilización de clústeres, es su tolerancia a fallos o caídas de servicio llamadas también “failovers”, lo cual significa que el o los servidores de respaldo toman el control de las tareas que estaba realizando el servidor que presento el fallo, y proceder a aislarlo del clúster para que no afecte al rendimiento hasta que sea analizada la razón por la cual fue remplazado. (Baker & Buyya, 1999) El uso de clústeres permite tener mayor seguridad al momento de realizar mantenimiento de hardware o de software mitigando en un 99% fallas en la continuidad del negocio.

En conclusión, los clústeres son usados para mejorar el rendimiento y/o la disponibilidad por encima de lo que provee por un único servidor obteniendo los siguientes beneficios:

- Aumentar la potencia y disponibilidad de los servidores.
- Facilita el aumento y balanceo de carga.
- Simplifica el manejo de recursos.

- Elimina los puntos de falla única del sistema.
- Permite brindar una operación del servicio del 99%.
- Redistribuye la carga de peticiones del servicio o aplicaciones para mejorar los tiempos de respuesta.

Los clústeres permiten a las organizaciones impulsar sus sistemas utilizando tecnología estándar que puede ser adquirida o comprada a un costo relativamente bajo. Con esto se provee una escalabilidad y mejoramiento a un costo accesible, que permite a las organizaciones potenciar a los ordenadores y a la vez preservar sus inversiones sin incurrir con muchos gastos adicionales.

1.1.2. Clasificación

La clasificación de los clústeres son establecidos de acuerdo al uso que se les vaya a dar y a los servicios que vayan a prestar.

1.1.2.1. Alto Rendimiento (High Performance)

Los clústeres en alto rendimiento, son aquellos que ejecutan tareas, las cuales necesitan una gran capacidad computacional o también grandes cantidades de memoria. Este tipo de clúster cumple con el objetivo principal de obtener un mejor rendimiento en la velocidad de proceso de datos. (Branch Bedoya, Ramos Corchado, Mesa Múnera, & Jiménez Mejía, 2009)

Este tipo de clúster está diseñado para dar altas prestaciones en cuanto se refiere a capacidad de cálculo ya que presentan las siguientes características:

- Dado que el tamaño del problema que se necesita resolver es muy grande es necesario dividir en sub-procesos
- Se realizan capacidades de cálculo de forma más rápida que utilizando un servidor de última generación a un costo muy alto
- El tiempo en el cual se debe de obtener un análisis del proceso realizado se requiere que sea menor.

El método de funcionamiento de un clúster de alto rendimiento es a través de la división de un problema en sub-problemas, para calcularlos en cada uno de los nodos del clúster de forma paralela. De esta forma se agiliza la velocidad de procesamiento y se tiene una velocidad de respuesta más alta.

1.1.2.2. Alta Eficiencia (High Throughput)

Son clústeres que tienen la finalidad de ejecutar el mayor número de tareas en el menor tiempo posible. (Escuela Politécnica Nacional, Julio) Existe la independencia de datos entre las tareas individuales de los servidores.

1.1.2.3. Alta Disponibilidad (High Availability)

Clústeres en alta disponibilidad, tienen como objetivo principal brindar disponibilidad y confiabilidad. Brindan la máxima disponibilidad en todos los servicios que se ofrecen en el área de soporte en infraestructura (24x7x365) (Branch Bedoya, Ramos Corchado, Mesa Múnera, & Jiménez Mejía, 2009), “la confiabilidad se la alcanza por medio de un software específico que está a cargo de detectar fallos y recuperarse lo más pronto posible”. (Escuela Politécnica Nacional, Julio)

“Funciona entre dos o más servidores que tienen servicios compartidos y se encuentran monitoreando constantemente entre sí.” (Clavijo, 2010) Esta clasificación de clústeres son los más demandados en el mercado por las empresas que desean asegurar el servicio a sus clientes.

La Alta Disponibilidad es el concepto que permite tener un fallo en uno de los nodos del clúster y el software se encarga de iniciar automáticamente los servicios en cualquiera de los otros nodos que están involucrados en el sistema del clúster. Cuando el nodo principal que genero el fallo se recupera, los servicios pueden ser nuevamente migrados al nodo original del clúster. Con esto es imperceptible para el cliente el fallo en el sistema.

1.1.3. Tipos de Implementación de clústeres

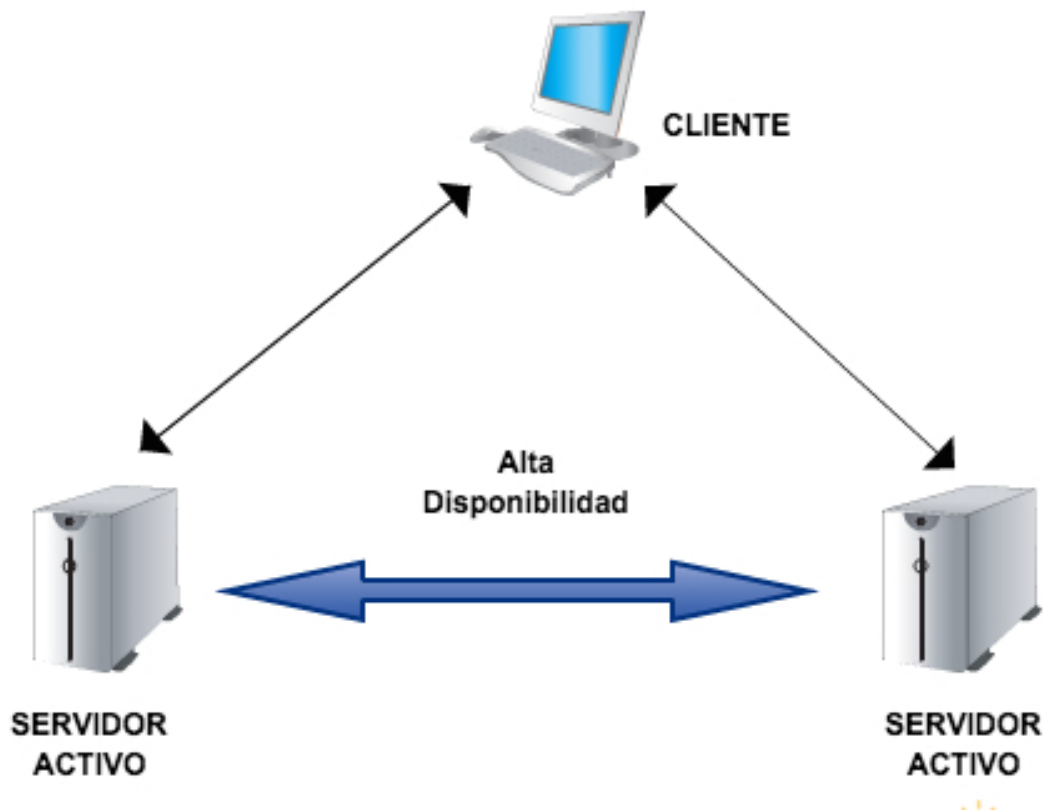
1.1.3.1. Clústeres Activo-Activo

En un clúster Activo-Activo todos los servidores pueden realizar las mismas tareas simultáneamente, esto quiere decir, que todos los servidores poseen las mismas características a nivel de Hardware y de Software. Si un nodo llega a fallar y no está disponible, sus recursos continúan accesibles a través de cualquiera de los otros nodos del clúster.

La ventaja de utilizar este tipo de implementación es que los nodos del clúster son más eficientes. Los clientes no tienen conocimiento de que existen varios servidores trabajando en el clúster.

En el gráfico siguiente se puede observar cómo trabaja un clúster activo-activo, los dos servidores proporcionan el mismo servicio a diferentes usuarios, los usuarios acceden al servicio de una forma transparente.

Figura 1 Clúster Activo-Activo



Fuente: www.lintips.com/?q=node/119

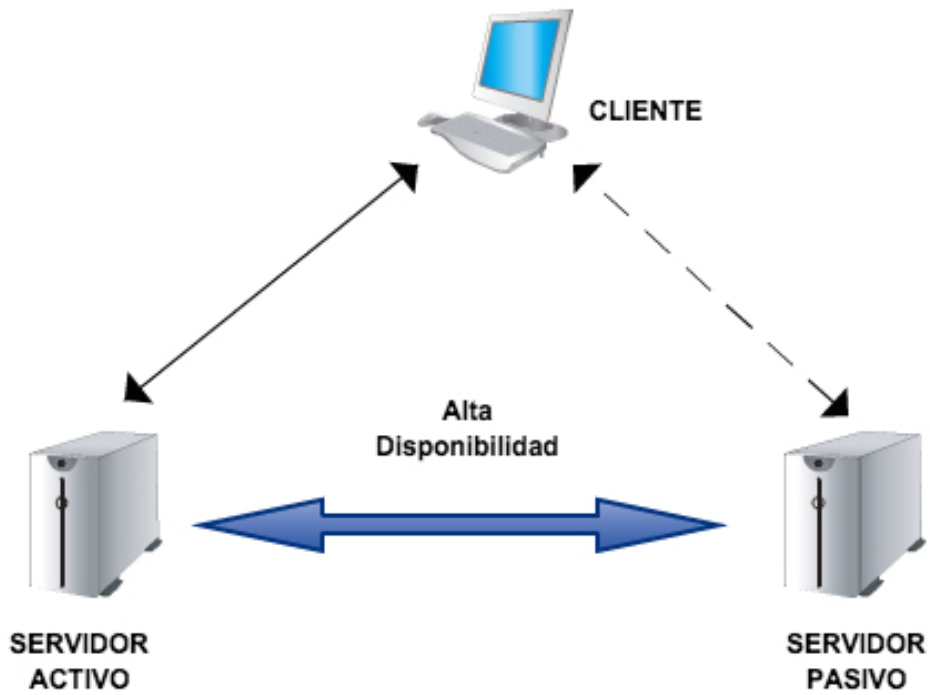
Realizado: Gabriel Andrade

1.1.3.2. Clústeres Activo-Pasivo

En esta configuración de clúster, solo un nodo es el que trabaja para brindar el servicio, lo que significa que solo este servidor es el encargado de los recursos del clúster mientras que los otros servidores se encuentran en espera de tomar el rol principal del clúster, y solo serán activados cuando el nodo principal deja de funcionar por alguna razón. La ventaja principal de este tipo de clúster es que no existe una degradación en el servicio y solo cuando el nodo activo presenta una falla se reinician los servicios. (Clavijo, 2010)

Los nodos que se encuentran pasivos no proporcionan ningún tipo de recurso mientras se encuentren en ese estado, lo cual hace que sea una solución menos eficiente que la anterior (activo-activo). Otra desventaja que presenta el clúster activo-pasivo es que en un momento que se presente una caída de servicio o conocido como failover, este tarda un tiempo en migrar los recursos a los demás nodos que se encuentran en espera.

Figura 2 Clúster Activo-Pasivo

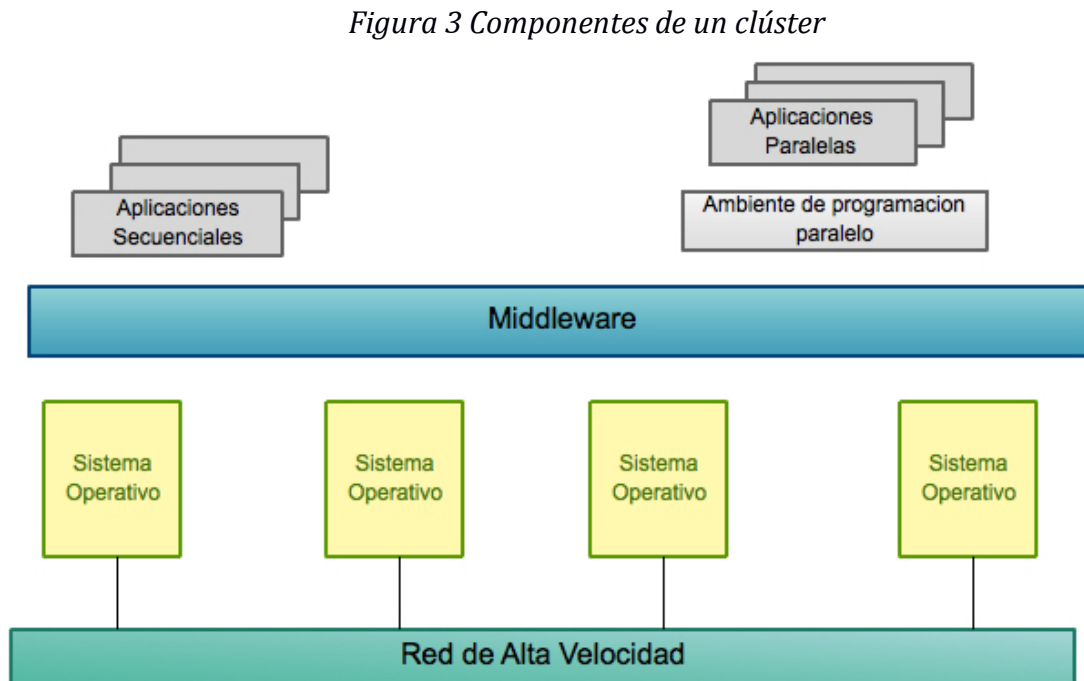


Fuente: www.lintips.com/?q=node/119

Realizado: Gabriel Andrade

1.1.4. Componentes de un Clúster

Los componentes y la arquitectura típica de un clúster están demostrados en la Figura 3.



Fuente: <http://clusterfie.epn.edu.ec/clusters/Definiciones/definiciones.html>

Realizado: Gabriel Andrade

1.1.4.1. Nodos

Es el nombre que se le da a cualquier servidor que se utiliza para montar el clúster. Un nodo puede ser un simple multiprocesador con memoria, Sistema Operativo, y con periféricos entrada/salida, tal como PC o estación de trabajo. (Escuela Politecnica Nacional, Julio) Se recomienda que todos los nodos tengan las mismas capacidades y características, ya que si no es así, se tendrá la tendencia de enviar el trabajo al nodo que tenga la mayor capacidad de procesamiento.

Los nodos que conforman el clúster tienen las siguientes características:

- Todos los nodos tienen acceso a los mismos datos de la configuración del clúster.
- Todos los nodos saben cuándo otro sistema se une o abandona al clúster.
- Los nodos son agrupados con un nombre específico en común, que es el nombre del clúster al que pertenecen.
- Para que los nodos se conecten entre sí deben estar en el mismo segmento de red, con esto se evita pérdida de paquetes.

1.1.4.2. Sistema Operativo

Se puede utilizar el mismo sistema operativo para los nodos de un clúster, en el desarrollo de esta guía metodológica se usará Red Hat Enterprise Linux 6.6. Además deben cumplir con dos características esenciales: deben ser multiproceso y multiusuario, ya que servidor necesita ejecutar varias tareas simultáneamente.

Para la gestión y administración de los clústeres en sistemas GNU Linux se tienen varias herramientas de gestión, la más conocida es la conocida como Conga, este involucra agentes en cada uno de los nodos que conforman el Clúster. Para el manejo del clúster que se desarrolla en esta guía se utiliza un clúster a nivel de aplicación conocido como mod_cluster.

Es recomendable que el administrador del clúster, tenga un nivel básico de experiencia con el manejo de sistemas GNU Linux, ya que la guía metodológica funciona bajo este sistema, por motivos seguridad, dado que cuenta con iptables, SELinux y políticas de seguridad para usuarios y grupo de usuarios.

1.1.4.3. Red de Interconexiones de altas prestaciones

Todos los nodos deben estar conectados entre sí, para ello se utiliza redes de Ethernet como FastEthernet (100Mbps), GigabitEthernet (1/10Gbps) u otros sistemas de alta velocidad tales , Myrinet (2/10Gbps), Infiniband (96Gbps), SCI (5,3Gbps), etc. (Perez F. , 2013)

En la actualidad las más utilizadas son las redes de Ethernet, ya que tienen un bajo costo pero una desventaja es que la velocidad de transmisión de la información puede limitar el rendimiento en los clústeres. GigabitEthernet es la que más se utiliza presentando mejor respuesta en el rendimiento de los equipos por la velocidad a la cual se encuentran interconectados.

1.1.4.4. Middleware

La capa media de aplicaciones o Middleware es el nombre que recibe el conjunto de software que se encuentra entre las aplicaciones y el Sistema Operativo y es el encargado de proveer al usuario la plataforma de aplicaciones para su uso.

El Middleware está encargado de la gestión global de todos los recursos de un clúster a través del conjunto de servidores conformados en el clúster, con esto se consigue optimizar los recursos y realizar operaciones como balanceo de carga, y tolerancia a fallos. También el middleware se encarga de la escalabilidad de un clúster, es decir, detecta nuevas máquinas y las añade al grupo en caso de que se requiera mayor demanda en el consumo de recursos por parte de las aplicaciones. (Yima, Fuentealba, &

Quiroz, 2012)

Es usado para soportar aplicaciones distribuidas, las que incluyen servidores web, servidores de aplicaciones, sistemas de gestión de contenido; se utiliza especialmente en tecnologías XML y SOAP (protocolo que define cómo dos objetos se pueden comunicar en procesos diferentes por medio de intercambio de datos XML), servicios web y arquitecturas orientadas a objetos. (Sosa, 2014)

El rol principal de un Middleware es ocultar la complejidad de las aplicaciones distribuidas y asistirles para que puedan interactuar entre sí. La capa media de aplicaciones proporciona los servicios que reducen al mínimo la duplicación de esfuerzos y mejora la comunicación entre aplicaciones. El Middleware presenta a los usuarios la ilusión de estar usando un único servidor de aplicaciones sin que ellos se den cuenta de la infraestructura que está montada debajo de la capa de aplicaciones, esta dependerá de la demanda de recursos que las aplicaciones demanden para su uso.

De la misma manera la capa media de aplicaciones simplifica el trabajo de los programadores en la generación de conexiones y plantillas de código base para la comunicación entre las bases de datos y los aplicativos que se tienen sobre la plataforma de aplicaciones. El middleware recibe las peticiones de trabajo entrante y redistribuye de la mejor manera para que se ejecuten lo más rápido posible y que no presente sobrecargas sobre los sistemas operativos base. Esto se lo realiza gracias a políticas que están definidas en la configuración del clúster.

Con el middleware se tiene alta disponibilidad de los servicios que están definidos en el clúster de aplicaciones, esto se lo consigue con redundancia de hardware. Una ventaja del uso de middleware es que permite la migración de los procesos entre los servidores del clúster para balancear la carga de las aplicaciones, y poder dar mantenimiento a los servidores físicos y dar priorización de trabajos de mantenimiento de la infraestructura.

Se encarga de proveer lo siguiente:

- **Balanceador de carga:** Cuando un servidor está con muchos procesos, se puede transferir parte de los procesos a otro servidor que no esté cargado de trabajo, con esto se optimiza el funcionamiento del clúster.
- **Mantenimiento a los servidores:** Cuando un servidor requiere actualización o mantenimiento, se migra los procesos a otro servidor del clúster y así se puede desconectar sin afectar el trabajo del clúster.

- **Priorización de trabajos:** Cuando se tiene procesos en un servidor pero uno de ellos requiere de una mayor importancia que los demás, se puede migrar dicho proceso a otro servidor que posea más recursos en Hardware para que los mismos sean realizados con mejor tiempo de respuesta.

1.2. Servidor de Aplicaciones

Un servidor de aplicaciones es conocido como la herramienta que proporciona el Hardware necesario en la infraestructura de servidores y servicios necesarios para el despliegue de aplicaciones, es decir, se lo puede llamar también como un contenedor de aplicaciones. (Universidad de Alicante, 2003) La mayor ventaja de utilizar Servidores de Aplicaciones es tener una configuración centralizada con esto se obtiene una disminución de tiempo en el desarrollo de aplicaciones, ya que el grupo de desarrolladores se preocupa solamente en el desarrollo de la aplicación. La tarea de administración del servidor donde se aloja la aplicación, se deja al grupo encargado del despliegue de la infraestructura necesaria sobre la cual se levantarán las aplicaciones, es decir, el desarrollador no se preocupa por las interfaces, las conexiones a la base de datos y de mantener la seguridad y control de usuarios.

Los servidores de aplicaciones están incluidos en capa media o middleware, que está encargado de poner todo el conjunto de aplicaciones en un solo sitio. Los servidores de aplicaciones proporcionan una Interfaz de Programación de Aplicaciones (API) a los desarrolladores.

Unas de las características principales de los servidores de aplicaciones son: (Universidad de Alicante, 2003)

- **Alta Disponibilidad:** Se refiere a que el sistema debe funcionar las 24 horas al día los 365 días del año.
- **Escalabilidad:** La capacidad de poder crecer con nuevos sistemas que se agreguen al conjunto del clúster cuando la carga de trabajo aumenta.
- **Mantenimiento:** Es la facilidad en el momento de actualizar, depurar fallos y poder mantener el sistema de la forma más sencilla.

Es muy importante tener clara la diferencia entre un servidor de aplicaciones y un servidor web. El servidor web funciona en dos capas mientras que el servidor de aplicaciones lo hace en tres capas lo cual permite estructurar al sistema de una forma

más simple y eficiente. A continuación se detallan las tres capas: (Universidad de Alicante, 2003)

- **Capa 1:** El navegador en el ordenador del usuario debe ser capaz de traducir el código del lado del cliente (HTML, JavaScript, CSS, Flash).
- **Capa 2:** La compone el servidor de aplicaciones, en la cual se debe traducir el código del lado del servidor y convertirlo a un formato que pueda ser entendible por el navegador del cliente.
- **Capa 3:** Están todos los servicios que accede el servidor de aplicaciones para poder realizar el trabajo que le ordenan (como por ejemplo el de acceso a la base de datos).

Con la utilización de servidores de aplicaciones se mantiene la integridad de datos y códigos, ya que al estar centralizada toda la información en una o en un pequeño grupo de máquinas que actúan como servidores, todas las actualizaciones están aseguradas para cada usuario que use el sistema. No existe el riesgo de utilización de versiones anteriores. Existe una configuración centralizada, ya que todos los cambios que se realice en las aplicaciones, pueden ser realizados centralmente.

Generalmente se suele confundir a Tomcat como un servidor de aplicaciones, pero Tomcat en realidad es solo un contenedor de servlets. La diferencia principal es que un contenedor de servlets está orientado únicamente para tecnología web, en cambio un servidor de aplicaciones es usado en sistemas complejos y en multicapa.

1.2.1. Servidor de Aplicaciones J2EE

Estos servidores de Aplicaciones utilizan la tecnología Java; la principal ventaja de utilizar J2EE, es que al ser basado en Java son multiplataforma, lo que significa que se puede utilizar la aplicación en todas las arquitecturas como Windows, Linux sin necesidad de realizar grandes cambios o configuraciones. El objetivo principal de la tecnología J2EE es la utilización de un modelo simple de desarrollo para aplicaciones empresariales; Java J2EE provee estándares que permiten a un servidor de aplicaciones actuar como un contenedor de todos los componentes de tales aplicaciones. (Oracle Corporation) En este modelo los componentes utilizan servicios que son proporcionados por el contenedor, que de no ser así tendrían que estar dentro del código fuente de las aplicaciones Los

componentes son desarrollados en código Java y se les conoce como Servlets, JSPs (Java Server Pages), y EJBs (Enterprise JavaBeans); estos permiten implementar las capas de las aplicaciones para la interfaz de usuario, la lógica del negocio, y acceso a las bases de datos remotas.

J2EE soporta aplicaciones distribuidas, en las que toma ventajas de tecnologías ya existentes; define estándares que son implementados por distintos proveedores y fabricantes, no fuerza a emplear ningún producto de software específico. Adicionalmente una de las principales ventajas de utilizar esta tecnología como plataforma es que se puede empezar con poco o en ciertos casos sin ningún costo, ya que existen muchas herramientas de código abierto para incrementar el tamaño de las plataformas para hacer más sencillo su desarrollo. (Zattola, 2014)

La arquitectura de un servidor de aplicaciones consiste en los siguientes subsistemas: (Universidad Autonoma de Baja California, 2004)

- Servidor HTTP: También es conocido como servidor web. Generalmente esta aplicado sobre TCP/IP; está basado en peticiones y respuestas que por defecto utiliza el puerto 8080, aunque se puede configurar para utilizar otro puerto.
- El Contenedor de Aplicaciones
- Un contenedor de Enterprise Java Beans, en el que se alojan todos los aplicativos java para la interacción con la Base de Datos o los sistemas empresariales

Todos estos subsistemas interactúan con el cliente de la siguiente forma el cliente que desea utilizar la aplicación, no la contacta directamente, primero se conecta con el servidor web por medio de HTTP. El servidor web es el intermediario en la solicitud; es el encargado de llevar la petición hasta el servidor de aplicaciones donde se aloja la aplicación requerida. El cliente por lo general a través de un explorador o una aplicación, pide un recurso por medio de HTTP, el cual es localizado especificando el URL (Localizador de Recursos Uniforme). (Oracle Corporation, 2014)

Los servidores de aplicaciones J2EE más comunes son Glassfish y JBoss, este último es en código abierto y es el más usado al momento por empresas en el sector público y privado.

1.2.1.1. JBOSS EAP (Enterprise Application Platform) 6.3

JBoss EAP 6.3 es un contenedor de aplicaciones que cumple con los requerimientos de la especificación Java empresarial edición 6.3; es una multiplataforma de código abierto basado en Java, la primera versión fue realizada por Marc Fleury, con la que creo JBoss Inc., pero posteriormente fue adquirida por Red Hat en abril del 2006. JBoss tiene su versión comunitaria, que cualquier persona puede descargarla desde la página oficial de JBoss, la desventaja de utilizar la versión comunitaria es que si en algún momento se presenta una falla, la solución depende de las respuestas que se encuentran en la comunidad. Para evitar esto Red Hat, tiene la versión empresarial, de esta forma los usuarios que utilizan JBoss tienen la seguridad de que cualquier problema que pueda surgir, son solucionados con soporte de Red Hat que está incluido con los tipos de suscripción disponibles 24 x 7 o 5 x 8.

JBoss EAP 6.3 posee una administración centralizada de múltiples instancias del servidor. Es una plataforma middleware, rápida y segura; provee clustering con alta disponibilidad, mensajería rápida, cache distribuido y otras tecnologías que la ayudan a ser una plataforma estable y escalable. La nueva estructura que presenta JBoss EAP 6 permite que los servicios se activen únicamente cuando sean necesarios, lo que permite que aumente significativamente la velocidad del trabajo. La consola de administración y la interfaz de línea de comandos (CLI) eliminan la necesidad de editar manualmente los archivos de configuración XML, con esto se añade la posibilidad de automatizar los procesos. En la versión de JBoss EAP 6 todos los archivos de configuración están centralizados, a diferencia de las anteriores versiones de JBoss EAP que había muchos archivos de configuración; con esta mejora se ahorra tiempo y es más fácil de realizar cambios en las configuraciones. JBoss EAP 6 soporta clustering en diferentes niveles, provee balance de carga y los beneficios de failover. (Red Hat, Inc., 2014)

1.2.1.2. Glassfish

Glassfish es un servidor de aplicaciones hecho en software libre, que utiliza el estándar J2EE, es desarrollado por Sun Microsystems, que pertenece a Oracle. (Oracle Corporation, 2013) Glassfish es distribuido a través de un licenciamiento dual, lo que significa que usa licencia CDDL (Common Development and Distribution License) y GNU GPL (General Public Licence). (Oracle Corporation, 2013) EL código fuente de Glassfish es donado por la Sun y Oracle Corporation.

1.2.2. Otros tipos de Servidores de Aplicaciones

Existen otros servidores de aplicaciones que no utilizan el estándar JEE y emplean otros lenguajes de programación que son utilizados en la actualidad, como .Net; .Net ha tenido un crecimiento en el Mercado, por lo que Microsoft decidió lanzar su propio servidor de aplicaciones llamado: Internet Information Server.

Así mismo, existen servidores de aplicaciones que utilizan código abierto como por ejemplo Base4 Server y Zope. Los cuales no son tan populares por no tener soporte oficial del fabricante y solo disponer información de foros, razón por la cual la tendencia a usar JBoss EAP 6 es la más usada en la actualidad por empresas privadas y gubernamentales.

1.3. Balanceador de Cargas

Un balanceador de cargas es una solución la cual permite dividir las tareas que son asignadas a un único servidor a otros servidores que encuentran en una misma red, esto se lo hace con el propósito de maximizar el proceso de datos y la ejecución de tareas. Con un balanceador de cargas se consigue que ningún equipo sea indispensable en el servicio que se desea ofrecer. (Palacio & Abril, 2012)

Tiene dos características principales:

- Evitar la saturación de un servidor. Se evita que los picos de acceso al servidor puedan afectar el desempeño de la aplicación.
- Se gestiona de una manera eficiente los recursos. Permite optimizar y gestionar todos los recursos para que de esta manera los aplicativos que se tienen implementados funcionen de la mejor forma posible.

1.3.1. Balanceo de Carga en Hardware

Los balanceadores de carga en Hardware son servidores con un propósito único, que es el de balanceo, no se puede utilizar a dichos servidores para algún otro propósito, lo cual es una gran desventaja ya que el servidor es subutilizado y aumenta el costo en el montaje de los datacenters de las organizaciones. No obstante al utilizar balanceo de carga por Hardware, solo para balanceo se tiene una gran potencia y estabilidad.

Así mismo cabe recalcar que tiene este tipo de balanceador es en cuestión de precio, son muy caros y el mantenimiento es extremadamente alto; funcionan como una caja cerrada, es decir, el desarrollador no puede manipular el código fuente ni cambiar la estructura; solo se lo administra por una interfaz web.

1.3.2. Balanceo de Carga en Software

Los balanceadores de carga en Software son servidores que son configurados para realizar la tarea de balanceo, para esto el servidor necesita tener un sistema operativo y una aplicación para que se encargue del balanceo. Estos se los conocen como balanceadores de carga en capa 4 del modelo OSI, balanceadores de carga a nivel de software. (Palacio & Abril, 2012)

Las ventajas de utilizar este tipo de balanceador es en dinero, ya que este balanceador es más barato que los balanceadores en hardware; adicionalmente se puede utilizar en cualquier servidor, y si se necesita una mayor potencia en balanceadores este servidor puede ser configurado para otra utilización.

1.3.2.1. Round Robin

Esta técnica de balanceo de carga funciona mediante DNS, de tal forma que cada vez que se realiza una petición al servidor que contiene en el registro DNS en cuestión, este contiene varias correspondencias del registro del servidor; de esta forma va rotando las peticiones entre los servidores, y técnicamente estaría balanceando al clúster (Conocimientos Informaticos). Este es un sistema muy sencillo de realizar pero es inconsistente, ya que si uno de los servidores a los que apunta deja de funcionar no hay como re-direccionar en un periodo rápido las peticiones que se realizaron dirigidos hacia el servidor que presento la falla.

Este tipo de balanceo solo se recomienda para páginas web que sean solo informativas o que no tengan interacción con el usuario, ya que cuando se almacena información como por ejemplo un carro de compras, la información es más importante de mantenerla en línea como en la página de Amazon y se decide hacer check out el usuario será redirigido a otro servidor en el clúster y la información del carro de compras se perderá, ya que la información se quedó almacenada en el anterior servidor. Lo cual generaría malestar a los usuarios y por ende no presentaría una plataforma estable para la organización.

1.3.2.2. Módulo mod_cluster

Es un balanceador de carga que es basado en httpd. Como mod_jdk, que se usaba anteriormente, mod_cluster utiliza un canal de comunicación para reenviar las solicitudes http a un conjunto de servidores que se encuentran en un clúster de servidores de aplicaciones. La diferencia principal entre mod_cluster con mod_jdk, es que mod_cluster aprovecha una conexión adicional entre los nodos del servidor de aplicaciones y httpd. Los nodos del servidor de aplicaciones usan esta conexión para transmitir información

del balanceador de cargas y eventos registrados a través de métodos HTTP, llamados mod_cluster Management Protocol (MCMP). Este canal usado para la retroalimentación permite a mod_cluster tener un nivel de inteligencia y granularidad para así poder mejorar continuamente. (jboss.org)

Mod_cluster trabaja en capa 7 en el modelo OSI lo cual significa que funciona a nivel de aplicaciones. Lo que hace es inspeccionar el mensaje hasta la capa 7, es decir hasta la petición HTTP, lo cual permite analizar la petición y sus encabezados, lo cual se usa como estrategia para el balanceo de carga. Con esto se puede realizar el balanceo en base a la cadena de la consulta, en cookies o en algún encabezado que se escoge

En contraste con los equilibradores de carga basada en httpd tradicionales, mod_cluster utiliza factores de equilibrio de carga que son calculados y proporcionados por los servidores de aplicaciones, en lugar de calcular estos datos en el proxy. En consecuencia, mod_cluster ofrece un conjunto más robusto y preciso de las estadísticas y factores de carga que está disponible en el proxy. Mod_cluster conoce cuando una aplicación no está desplegada así que no envía solicitudes para su contexto (por medio del URL) hasta que esta esté desplegada. (jboss.org)

1.3.2.3. Linux Virtual Server (LVS)

Linux Virtual Server es una alternativa para realizar balanceo de carga en servidores Linux. Con la utilización de LVS se puede proporcionar alta disponibilidad, alto rendimiento y confiabilidad en los sistemas que utilicen clustering. LVS aporta como una solución de alta disponibilidad a servicios de red que pueden ser Web, ftp, cache, mail, medios de comunicación. Este servidor permite realizar un balanceador de carga capa 4, en el modelo OSI, lo cual significa que es a nivel de TCP/IP; que funciona por ejemplo cuando viene una petición de un cliente en el puerto 80 y lo que hace el balanceador de carga es distribuir esta petición en los servidores que estén funcionando en el clúster en el mismo puerto que vino la petición. (linuxvirtualserver.org, 2011)

LVS funciona con un servidor director, que es el encargado de controlar al resto de servidores en el clúster. Este servidor director es el encargado de dirigir la carga de trabajo, en función de las reglas que se hayan configurado, al resto de servidores.

La ventaja de la utilización de LVS es que tiene un esquema activo-pasivo.

CAPITULO 2

2. CASO DE ESTUDIO SOPORTE LIBRE

2.1. Historia

Soporte Libre Cía. Ltda. es una empresa Ecuatoriana que cuenta con una trayectoria de más de 15 años en la industria, especializándose los últimos 8 años en herramientas de Software Libre y Open Source. Su misión es ofrecer soporte y capacitación de alta calidad en herramientas de Software Libre y Open Source a nivel nacional. La empresa Soporte Libre es Advance Business Partner de Red Hat y JBoss desde 2007 y Business Partner de Untangle desde 2009. Esta empresa de infraestructura, tiene como objetivo brindar a sus clientes servicios especializados sobre plataformas de Open Source y Software Libre. Maneja la visión de difundir entre las organizaciones las ventajas en el proceso de adopción de tecnologías FLOSS¹, además de apoyar los procesos de migración de las mismas en un ambiente controlado y soportado por el personal técnico de la empresa, utilizando las mejores prácticas reconocidas a nivel mundial basadas en la aplicación de conocimiento, procesos, metodologías, modelos, técnicas y experiencia.

Soporte Libre Cía. Ltda. ofrece servicios que se fundamentan en el conocimiento técnico/funcional y en la experiencia en proyectos del equipo de trabajo, los cuales ocupan distintos roles a todo nivel, además de valores éticos. Cada uno de los colaboradores ha participado en proyectos de desarrollo e implantación de FLOSS de mediana y gran escala tanto en Ecuador como en los Estados Unidos, realizando tareas diversas desde instalación, configuración y desarrollo hasta la gestión y administración de los proyectos. (Soporte Libre Cía. Ltda.)

MISION

Ayudar a las industrias locales para sacar el máximo provecho de la innovación aportada por la fuente abierta y el modelo de software libre, reduciendo sus costos, por lo tanto el aumento de nuevas inversiones en la educación para los empleados y otras mejoras tecnológicas. (Soporte Libre Cía. Ltda.)

VISION

Ser una empresa con alto prestigio y reconocimiento nacional, con servicios especializados sobre plataformas de Open Source y Software Libre, donde se brinde un producto y servicios de excelente calidad, en donde el mejoramiento continuo en todas las áreas sean de agrado a nuestros clientes, de eficiente gestión, competitiva, comprometida con el servicio al cliente y la formación integral de su recurso humano y

tecnológico. (Soporte Libre Cía. Ltda.)

LINEAS DE NEGOCIO (Soporte Libre Cía. Ltda, 2015)

Soporte Libre ofrece las siguientes líneas de negocio:

➤ **Sistema Operativo Linux**

Diseño, instalación, configuración y puesta en marcha de productos Red Hat Enterprise Linux, consolidación de servicios y servidores en infraestructura optimizadas.

➤ **Middleware Java**

Diseño, instalación, configuración y puesta en marcha de arquitecturas empresariales de software de capa media utilizando productos JBoss (servidor de aplicaciones, buses empresariales, reglas de negocio, portales, servicios de integración, servidor de publicación y balanceo de carga, servicios de integración).

➤ **Infraestructura para Nubes Publicas/Privadas**

Diseño, instalación, configuración y puesta en marcha de infraestructura para soluciones basadas en la nube y apegadas a los paradigmas IaaS o PaaS:

RHEV

OpenStack

OpenShift

➤ **Servidores Firewall**

Untangle Network Gateway.

➤ **Consultoría**

Definición de Arquitectura, migración a Software Libre, RFP de aplicaciones basadas en Open Source.

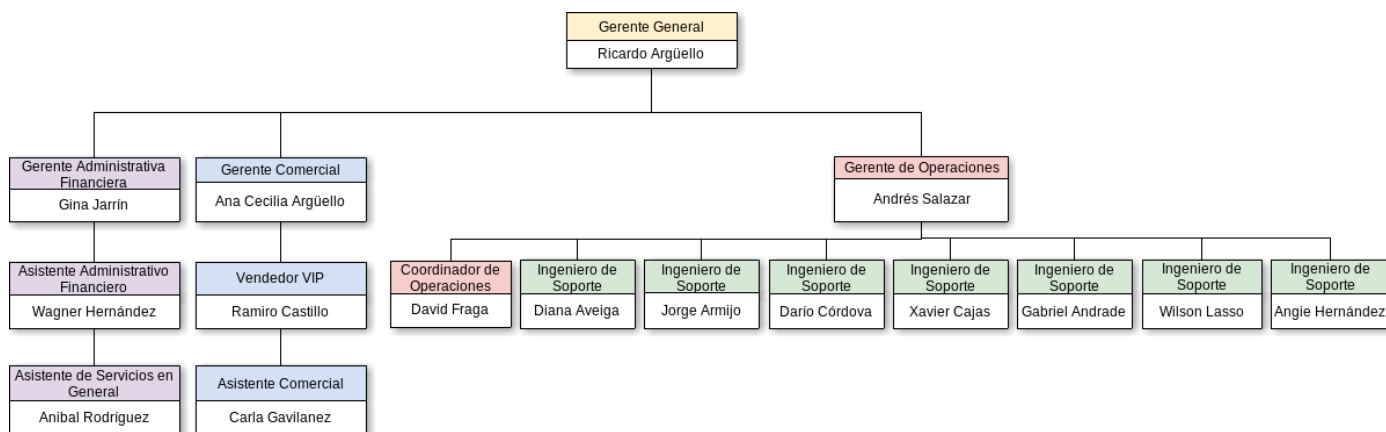
2.1. Organigrama (Arguello, 2015)

La empresa Soporte Libre Cía. Ltda. Está conformada por cuatro gerencias:

- **Gerente General:** Es conformada por el presidente de la empresa Ricardo Argüello. Especialista en infraestructura y desarrollo con más de quince años desarrollando proyectos y consultorías para el sector público y privado.

- **Gerencia Comercial:** Es conformada por el grupo de especialistas encabezado por Ana Argüello con más de veinte años en la gestión y venta de proyectos involucrados en la tecnología.
- **Gerencia Administrativa Financiera:** Es conformada por un grupo de expertos en la administración de empresas. Encabezada por Gina Jarrín.
- **Gerencia de Operaciones:** Es conformada por un equipo de ingenieros especialistas en tecnología OpenSource en el área de infraestructura, redes y middleware.

Figura 4 Organigrama Institucional Soporte Libre Cía. Ltda



Fuente: Soporte Libre Cía. Ltda

2.2. Mapa de Procesos (Arguello A. , 2015)

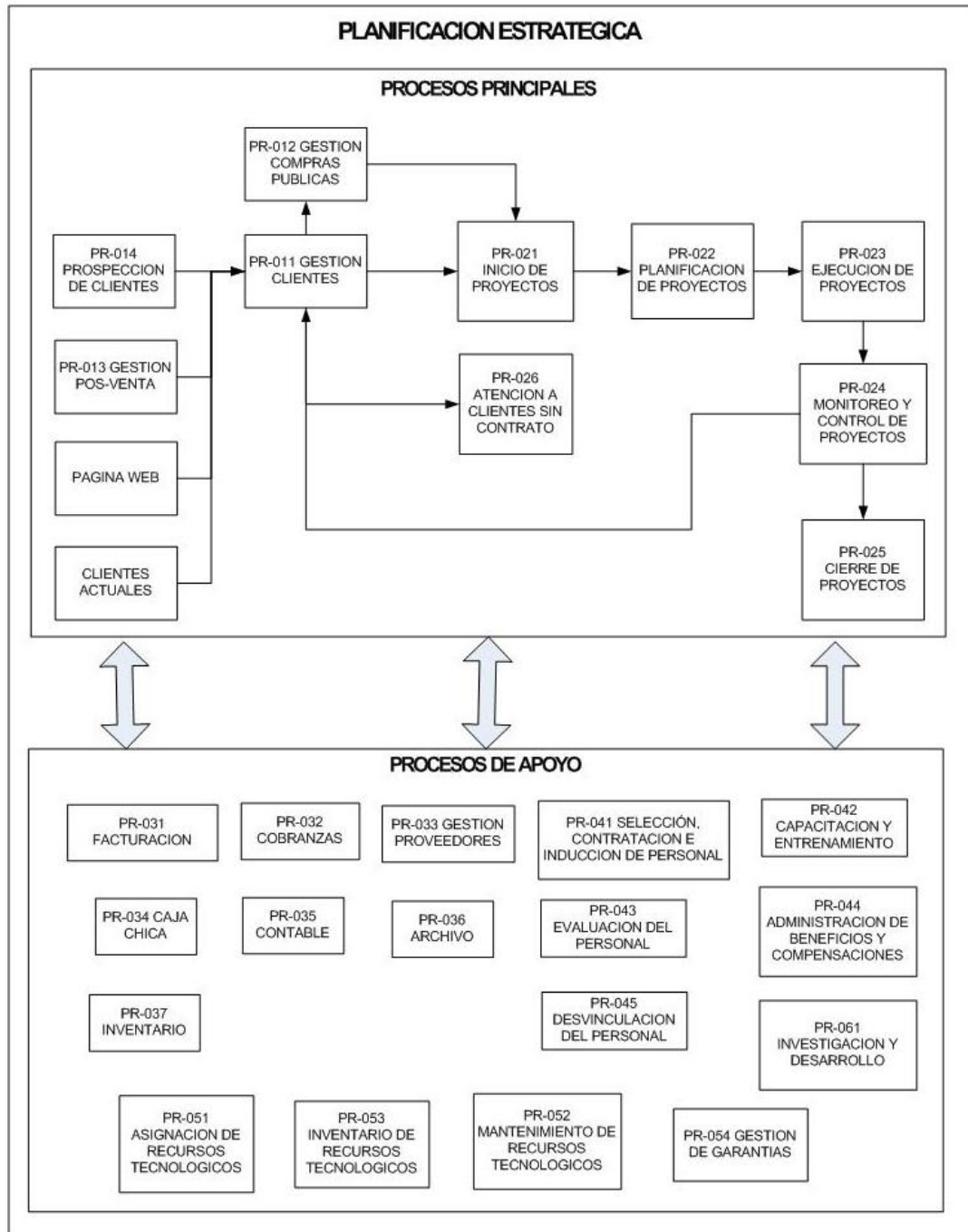
Los procesos en la empresa Soporte Libre Cía. Ltda. inician en la gerencia comercial con un listado de productos y servicios que son ofertados de acuerdo a los requerimientos en los clientes del sector empresarial privado y público. Una vez realizado el trabajo comercial en identificar las necesidades de cada cliente, las mismas son transmitidas a la gerencia de operaciones y a la gerencia administrativa para su respectivo análisis de ejecución tanto en el ámbito técnico como en el administrativo.

Una vez realizado este proceso, dependiendo del tipo de empresa, se procede al trámite legal respectivo para la adjudicación del proyecto. Una vez adjudicado el proyecto se continúa con la planificación y ejecución de las tareas involucradas. El ciclo del proyecto finaliza con un posterior análisis del área comercial en la cual se realiza un seguimiento

GUÍA METODOLÓGICA PARA LA IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA
DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES, UTILIZANDO OPEN SOURCE.
CASO DE ESTUDIO: EMPRESA SOPORTE LIBRE

del proyecto implementado y de los posibles requerimientos posteriores como resultado del proyecto realizado.

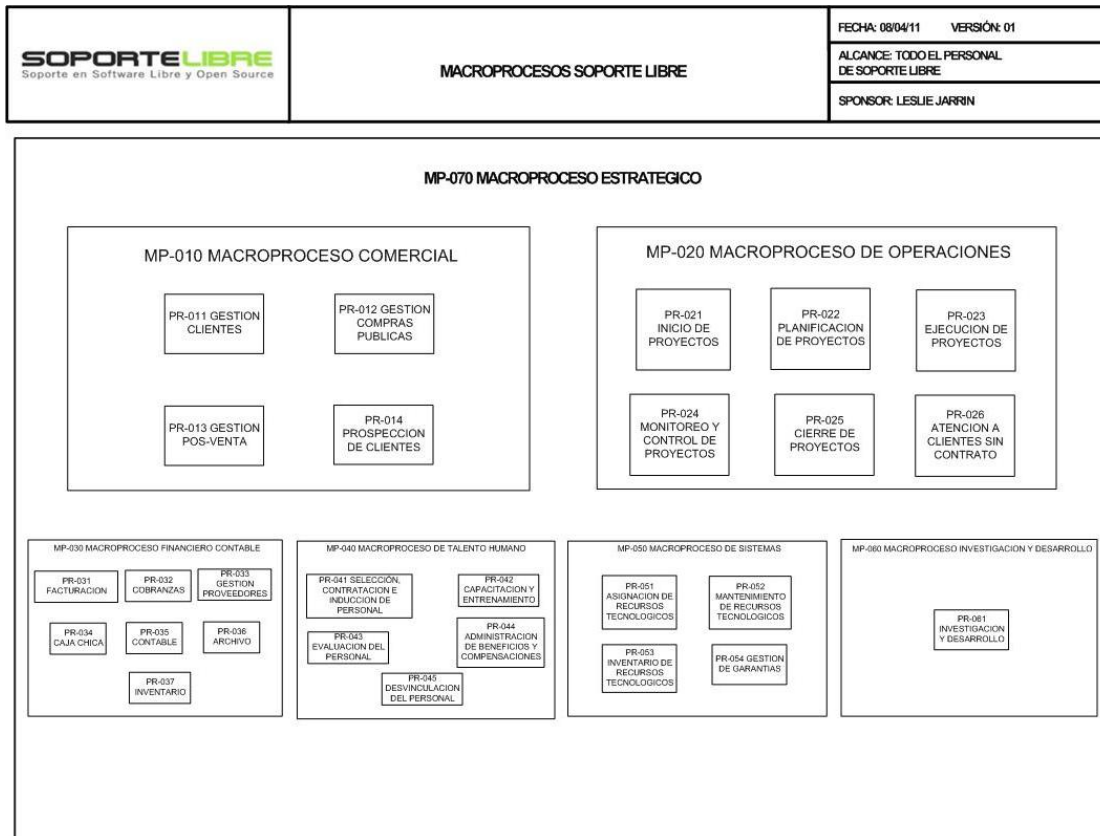
Figura 5 Planificación Estratégica Soporte Libre Cía. Ltda



Fuente: Soporte Libre Cía. Ltda

**GUÍA METODOLÓGICA PARA LA IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA
DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES, UTILIZANDO OPEN SOURCE.
CASO DE ESTUDIO: EMPRESA SOPORTE LIBRE**

Figura 6 Planificación Estratégica Soporte Libre Cía. Ltda



Fuente: Soporte Libre Cía. Ltda

2.3. Necesidad de Alta Disponibilidad

La necesidad de aumentar la alta disponibilidad en los sistemas utilizados por varias empresas continúa acelerándose debido a que muchos de estos usuarios siempre están rediseñando sus soluciones para conseguir una ventaja más competitiva en el mercado. Un acceso rápido a los datos críticos es de vital importancia, ya que si estos no están disponibles la aplicación dejaría de estar en actividad o hay una caída de sistema, lo cual conduce a una pérdida de productividad, e ingresos ya que existe un costo de inactividad que puede crecer de manera acelerada en las empresas que dependen de los servicios que prestan estas aplicaciones. Una caída del sistema no solo afecta al usuario, pero también a la empresa que diseño la aplicación ya que crearía una relación de desconfianza y de mala publicidad, teniendo graves implicaciones que hasta pueden resultar en acciones legales debido a que no se alcanza el servicio garantizado.

Soporte Libre Cia. Ltda., al ser una empresa con gran prestigio en el territorio nacional y con clientes que dependen inmensamente de los servicios que esta empresa provee, necesita ofrecer aplicaciones que protejan la integridad de los datos y que estos estén

GUÍA METODOLÓGICA PARA LA IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA
DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES, UTILIZANDO OPEN SOURCE.
CASO DE ESTUDIO: EMPRESA SOPORTE LIBRE

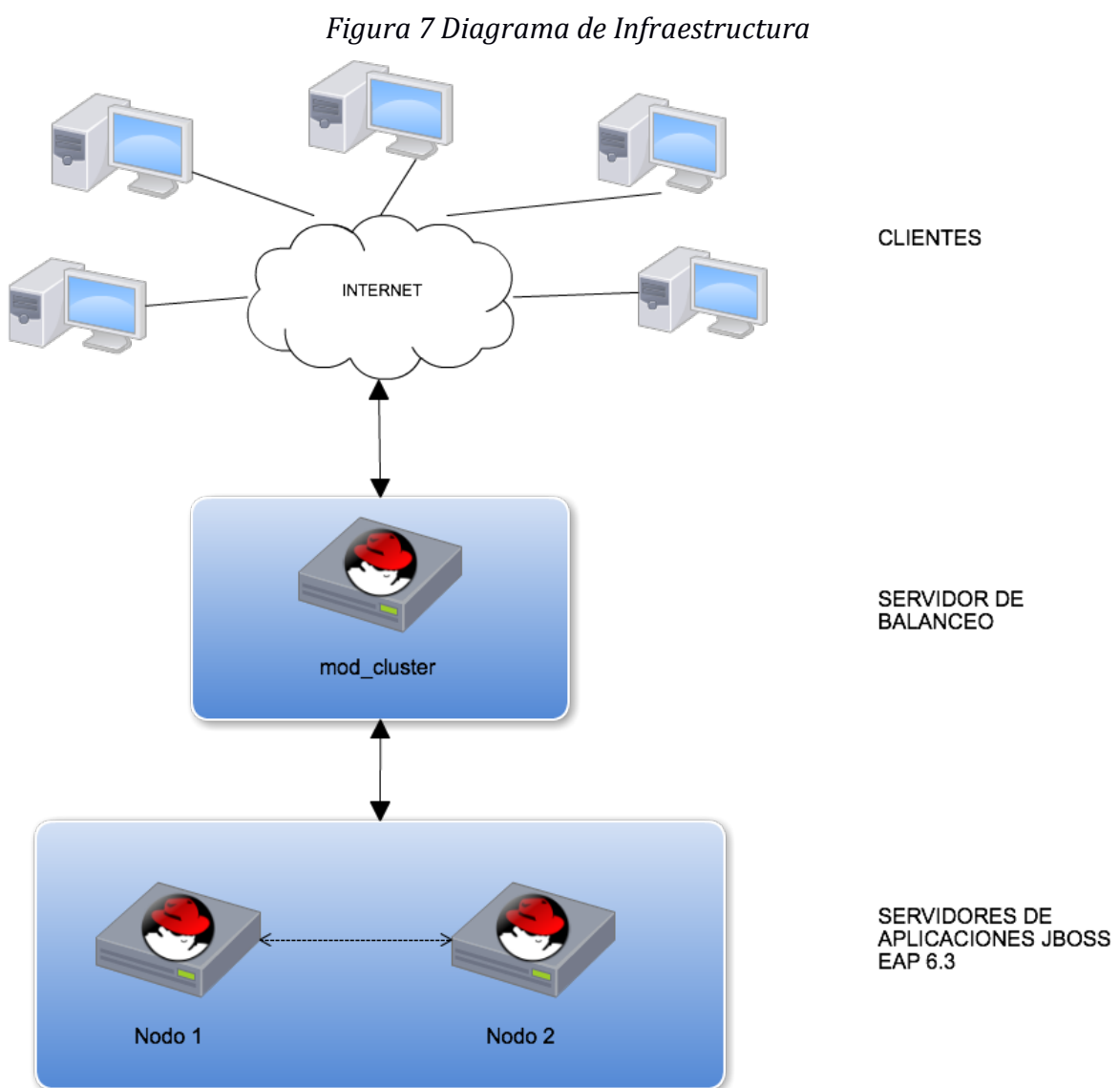
accesibles en todas partes y a cualquier momento. Es por esto que se recomienda que se implemente soluciones con alta disponibilidad que estén al alcance de todos sus clientes, independiente del tamaño de la empresa. Además de asegurar el funcionamiento continuo de la aplicación, con alta disponibilidad también se puede examinar y responder todas las posibles causas de una inactividad, desde un pequeño corte de luz a situaciones que hace la que duración de la inactividad sea mayor.

Para facilitar la implementación de un clúster en alta disponibilidad para servidores de aplicaciones por parte de Soporte Libre Cia.Ltda, se propone la creación de una guía metodológica, que permitirá tener una ruta estándar la cual se puede replicar en todos los clientes de la empresa, y que al ser validada se asegura su funcionamiento, facilitando el trabajo de los desarrolladores y proporcionando confianza hacia los clientes.

CAPITULO 3

3. GUIA METODOLOGICA PARA LA IMPLEMENTACION DE UN CLUSTER EN ALTA DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES

La siguiente es la infraestructura configurada para el desarrollo de esta guía metodológica:



Fuente: Gabriel Andrade

3.1. Configuración de Servidor

Para la instalación y configuración del Sistema Operativo de esta guía metodológica se escogió por Red Hat Enterprise Linux 6.6 x86_64 o también llamada RHEL6.6_x86_64 ya que es un Sistema Operativo que presta con todos los requerimientos que se necesita para un clúster.

Para instalar RHEL 6.6 se debe tener en consideración la plataforma en la cual se va a realizar la instalación. Se necesita un entorno de alta disponibilidad por lo cual se usará la plataforma de virtualización llamada Kernel-based Virtual Machine (K.V.M.). Este entorno va a permitir la representar un entorno virtual similar a un centro de datos físico.

Como primer paso para la realización de esta guía metodológica se necesitara la configuración de los siguientes componentes:

- a) Máquinas Virtuales para el entorno de un clúster de Alta Disponibilidad

Tabla 1 Equipos Virtuales para el entorno de un clúster en Alta Disponibilidad

No	Nombre	Descripción
1	jboss-nodo1	Nodo 1 de JBoss
2	jboss-nodo2	Nodo 2 de JBoss
3	Apache	Distribución de carga

- b) Configuración de red para las máquinas en el entorno de un clúster de Alta Disponibilidad

Tabla 2 Configuración de red para las máquinas en el entorno de un clúster de Alta Disponibilidad

No	Nombre	Hostname	Dirección IP	Máscara de Red	Puerta de Enlace
1	jboss-nodo1	jbnodo1.gabriel.com	192.168.9.5	255.255.0.0	192.168.9.1
2	Jboss-nodo2	jbnodo2.gabriel.com	192.168.9.6	255.255.0.0	192.168.9.1
3	Apache	apache.gabriel.com	192.168.9.7	255.255.0.0	192.168.9.1

3.1.1. Requisitos Mínimos de Servidor para soportar RHEL (Red Hat Enterprise Linux)

Antes de proceder a instalar el Sistema Operativo Red Hat 6.6 se debe tener en consideración el espacio de disco que se va a necesitar para cada máquina virtual.

En esta guía se va a necesitar un espacio menor de disco para la instalación y la configuración que en un ambiente en producción, ya que estas máquinas son de demostración. Así mismo para las dos máquinas que van a servir de nodos para JBoss se utilizará 20GB que están distribuidas de la siguiente forma:

Particionamiento del Sistema Operativo para Nodo de JBoss

Tabla 3 Particionamiento del Sistema Operativo para Nodo de JBoss

Punto de Montaje	Nombre	Tamaño en GB
/boot	boot	500 MB
/	lv_root	5 GB
/swap	swap	2GB
/opt	lv_opt	10 GB
/var	lv_home	1.5 GB
/tmp	lv_tmp	1 GB

Para la máquina que se va a utilizar para el Apache se va a necesitar 10 GB que estará distribuido de la siguiente manera:

Particionamiento del Sistema Operativo para Apache

Tabla 4 Particionamiento del Sistema Operativo para Apache

Punto de Montaje	Nombre	Tamaño en GB
/boot	boot	500 MB
/	lv_root	5 GB
/swap	swap	2 GB
/var/log	lv_var_log	1.5 GB
/tmp	lv_tmp	1 GB

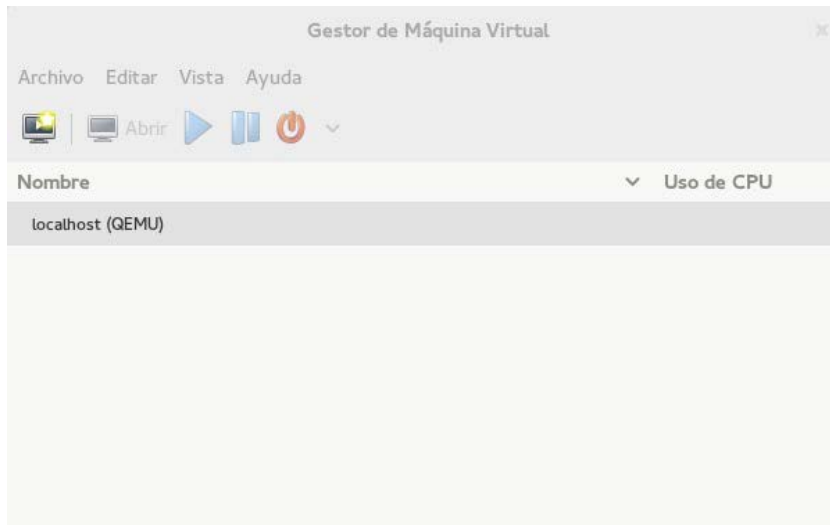
3.1.2. Configuración plataforma virtualización K.V.M.

Para la configuración del ambiente de virtualización que permitirá la instalación del esquema de clúster de JBoss, se deben realizar las siguientes configuraciones.

- a) Para acceder a la administración del K.V.M se debe ejecutar en la línea de comandos.

```
# virt-manager
```

Figura 8 Configuración Plataforma virtualización KVM Nro.1



Fuente: Resultado de configuración de plataforma KVM

- b) En el menú principal se debe seleccionar el icono para la creación de nuevas máquinas virtuales

Figura 9 Configuración Plataforma virtualización KVM Nro. 2



Fuente: Resultado de configuración de plataforma KVM

- c) Al presionar el ícono se va a abrir una nueva ventana para la creación de la máquina virtual. El primer paso es seleccionar el método por el cual se va a instalar el sistema operativo, para esta guía se va a utilizar una imagen de RHEL 6.6, por lo que se seleccionó la opción (imagen ISO o CDROM).

Figura 10 Configuración Plataforma virtualización KVM Nro. 3



Fuente: Resultado de configuración de plataforma KVM

3.1.2.1. Selección de medio de instalación

En esta guía se va a utilizar la imagen ISO de Sistema Operativo RHEL 6.6, que tiene como nombre `rhel-server-6.6-x86_64`, en caso de que no se disponga de una imagen ISO se puede utilizar también una unidad de DVD.

El tipo de Sistema Operativo se escoge “Linux” y en versión “Red Hat Enterprise Linux 6”

Figura 11 Selección de Medio de Instalación



Fuente: Resultado de configuración de plataforma KVM

3.1.2.2. Asignación de Memoria

Para poder asignar memoria a la máquina virtual, se debe tener en cuenta las características físicas del equipo en donde se está configurando el entorno de alta disponibilidad con JBoss, ya que de la máquina física depende la memoria RAM que va a utilizar la máquina virtual.

Lo recomendable es asignar 4096 GB de memoria RAM.

Figura 12 Asignación de Memoria



Fuente: Resultado de configuración de plataforma KVM

3.1.2.3. Asignación de espacio de almacenamiento

Para el caso de esta guía, que es un ambiente de pruebas, se asignará 20 GB para el disco duro de la máquina virtualizada que se usará como nodo y 10 GB para el disco duro de la máquina virtualizada que se usará como balanceador.

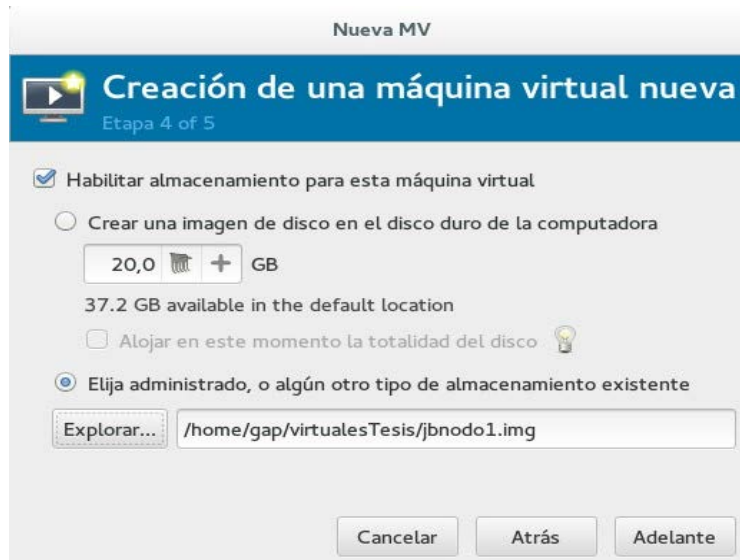
Se continúa a reservar el espacio de disco necesitado y la ruta en donde se almacenará.

Existen dos tipos de almacenamiento:

- Reservado dinámicamente
- Tamaño fijo

En el caso de esta guía se seleccionará “Tamaño fijo”, para que se reserve el total de espacio necesitado.

Figura 13 Asignación de Espacio de Almacenamiento



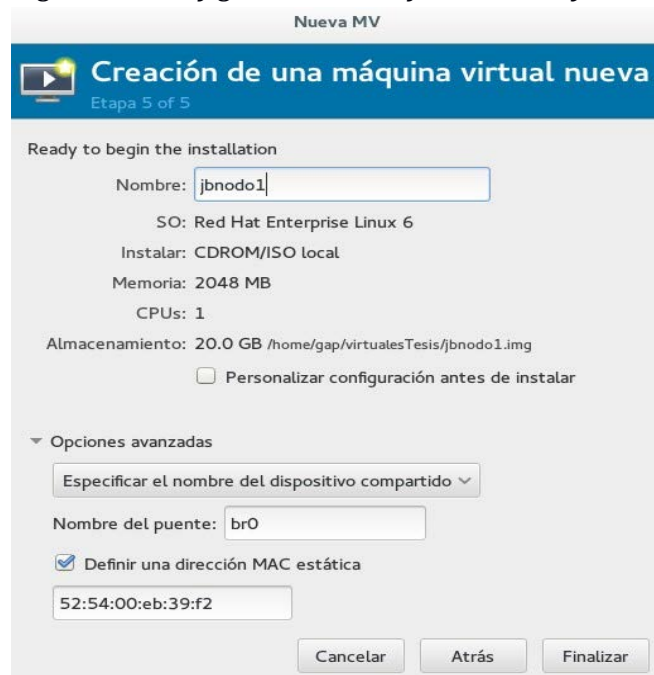
Fuente: Resultado de configuración de plataforma KVM

3.1.2.4. Configuración interfaces de red y nombre

Se debe definir el nombre que va a llevar la máquina virtual, lo recomendable es dar un nombre que describa el propósito de esta nueva máquina virtual.

También se procede a escoger la interfaz de red con la cual se va a tener acceso a los recursos compartidos. En el caso de esta guía se procede a elegir “Especificar el nombre del dispositivo compartido”; aquí se debe especificar el medio por el cual se va a tener acceso a la red interna

Figura 14 Configuración interfaces de red y nombre



Fuente: Resultado de configuración de plataforma KVM

3.1.3. Instalación de Sistema Operativo RHEL (Red Hat Enterprise Linux)

Una vez que ya configurado el Disco y la interfaz de red se procede a la instalación del Sistema Operativo Red Hat Enterprise Linux 6.6.

3.1.3.1. Menú de Arranque

El menú de arranque es un menú gráfico con varias opciones para el tipo de instalación. Las opciones de arranque son:

➤ **Install or upgrade an existing system**

Esta es la opción predeterminada. Es una instalación de Red Hat Enterprise Linux con un modo gráfico.

➤ **Install system with basic video driver**

En esta opción se instala Red Hat Enterprise Linux con modo gráfico, aunque el programa de instalación no pueda cargar los drivers correctos para la tarjeta de video.

➤ **Rescue installed system**

Se escoge esta opción para reparar, cuando el sistema Red Hat Enterprise Linux presenta problemas y no permite arrancar normalmente. Esta opción viene con

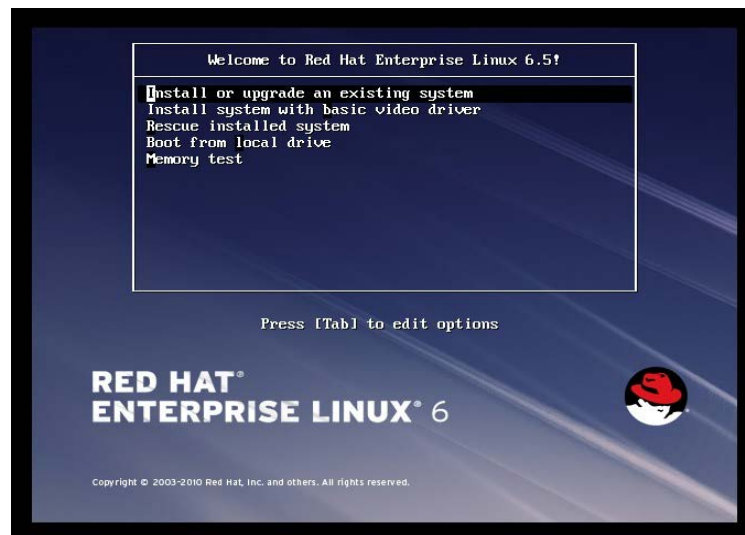
herramientas que permiten reparar y recuperar el sistema Red Hat Enterprise Linux que esté instalado.

➤ **Boot from local drive**

Se utiliza para levantar el Sistema Operativo desde un disco duro local

Para esta guía se selecciona la opción "Install or upgrade an existing system"

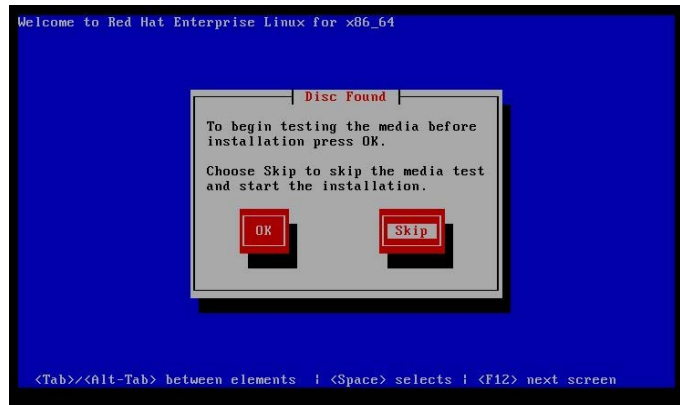
Figura 15 Menú de Arranque



Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6

Se tiene la opción de realizar una comprobación del medio de instalación, esto es para evitar que se instale el Sistema Operativo con errores; como se está usando una imagen ISO de rhel-server-6.6-x86_64 no requiere comprobación.

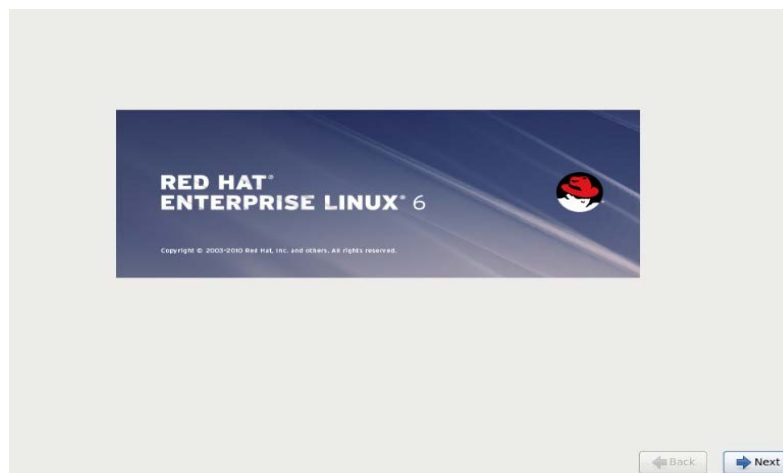
Figura 16 Comprobación Medio de Instalación



Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6

A continuación se procede con el asistente de instalación que se llama "Anaconda"; este ayudara a realizar las configuraciones del Sistema Operativo

Figura 17 Inicio de Instalación

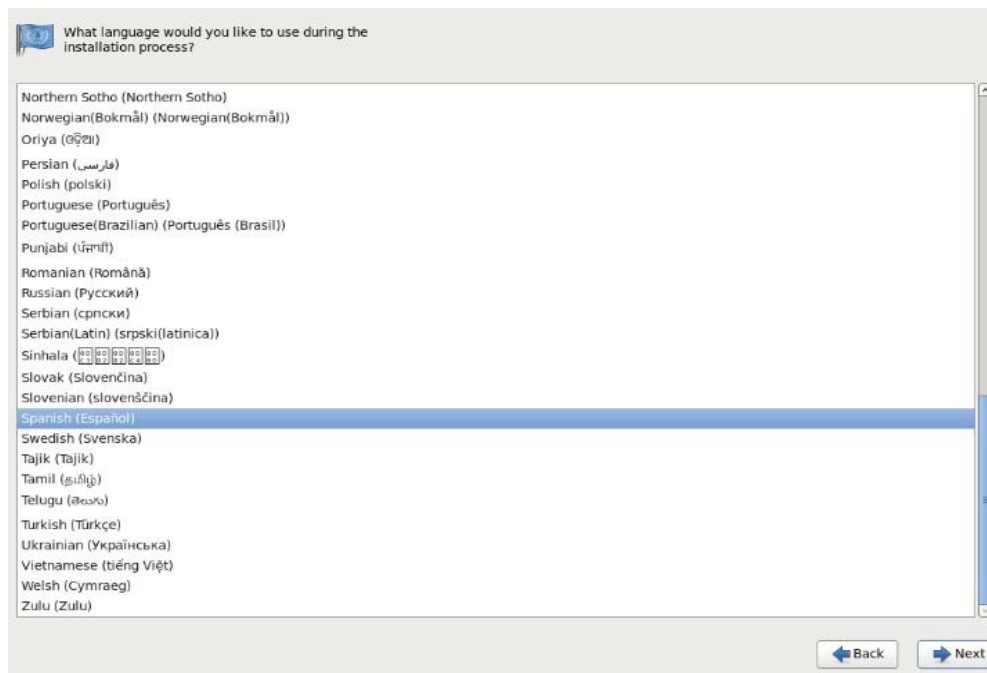


Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6

3.1.3.2. Selección de Idioma y teclado

Se selecciona el idioma de preferencia (en este caso Español); este idioma se va a utilizar para la instalación y también como predeterminado del sistema.

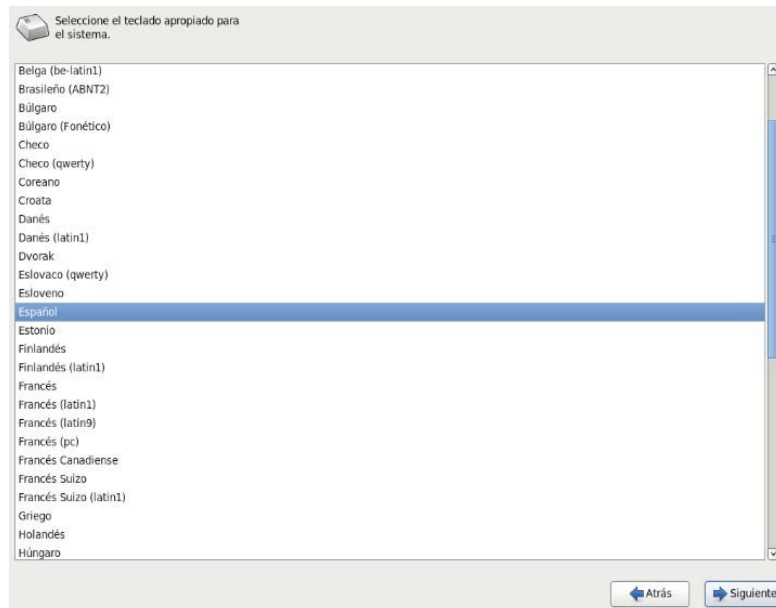
Figura 18 Inicio de Instalación



Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6

A continuación se selecciona el idioma del teclado apropiado (en este caso se usa Español) para la instalación y también como predeterminado del sistema.

Figura 19 Selección de teclado



Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6

3.1.3.3. Dispositivos de Almacenamiento

Se selecciona la opción que se vaya a utilizar para el medio de instalación. Se tiene dos tipos de dispositivos de almacenamientos que se pueden usar, que son:

➤ Dispositivos de Almacenamiento Básicos

Esta opción es utilizada para instalar Red Hat Enterprise Linux en discos duros que se encuentran conectadas al sistema local. Si se escoge esta opción, Anaconda reconocerá automáticamente el almacenamiento conectado al sistema, y no se requerirá más especificaciones.

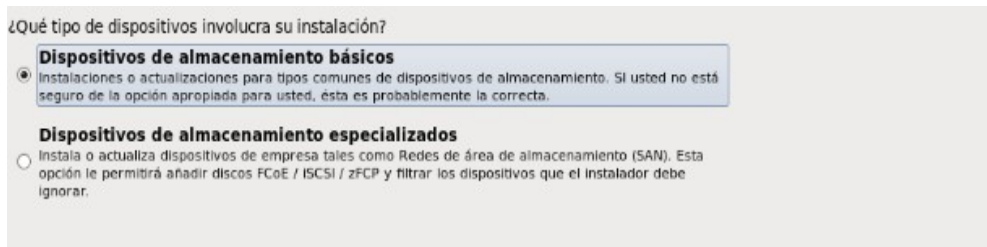
➤ Dispositivos de Almacenamiento Especializado

Se utiliza para instalar Red Hat Enterprise Linux en dispositivos de almacenamiento tipo SAN (red de área de almacenamiento), dispositivos RAID de Firmware, DASD (dispositivo de almacenamiento de acceso directo)

Para esta guía se va a escoger la opción de “Dispositivos de Almacenamiento Básico”. A continuación se va a presentar una advertencia en la que se indica que el disco

seleccionado se va a formatear y que toda información que contenga dicho disco se va a perder. Se procede a descartar

Figura 20 Dispositivos de Almacenamiento Parte 1



Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6

Figura 21 Dispositivos de Almacenamiento Parte 2

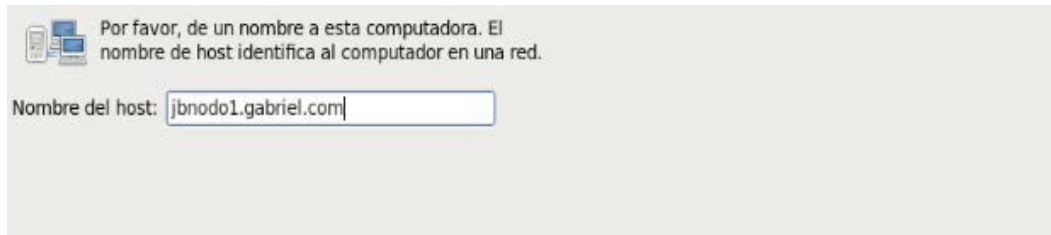


Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6

3.1.3.4. Configuración de nombre de host

El nombre del host es el nombre con el cual será identificado en la red el servidor. Es recomendable que este nombre tenga la sintaxis de un nombre de dominio totalmente calificado (FQDN), en el cual se incluye el nombre del servidor y el dominio con el que esta asociado.

Figura 22 Configuración de nombre de host



Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6

La configuración de red se la va a realizar al terminar la instalación del sistema operativo.

3.1.3.5. Configuración de huso horario

Se configura el huso horario seleccionando la ciudad más cercana que se encuentra a la ubicación física del servidor.

Existen dos formas para seleccionar el huso horario:

- Se puede seleccionar la ciudad con el huso horario que se necesite por medio del mapa interactivo. Una X de color rojo indica la selección que se ha tomado.
- Existe una lista desplegable en la cual se escoge la ciudad con el huso horario que se requiera.

Figura 23 Configuración de huso horario

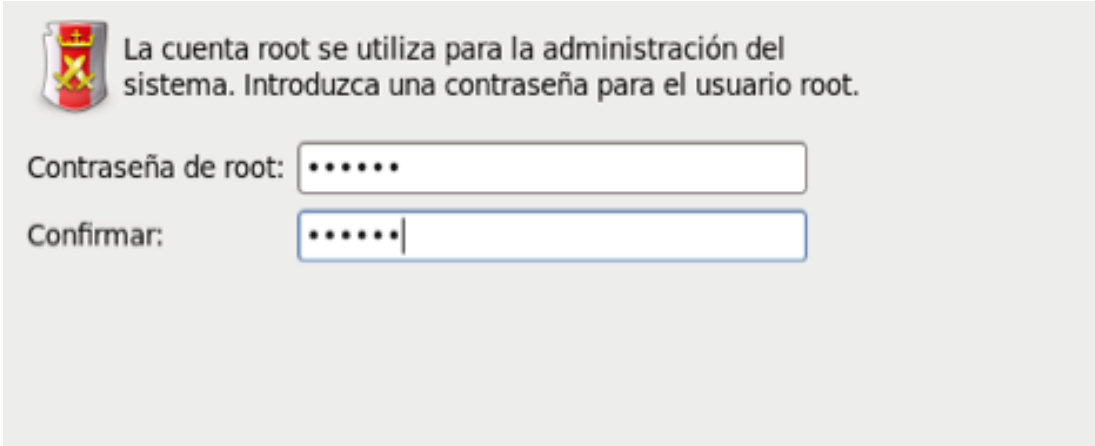


Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6

3.1.3.6. Establecimiento de contraseña de usuario root

El usuario root, que también se le conoce como super-usuario, tiene acceso a todo el sistema. EL root es el encargado de instalar paquetes, actualizar RPMs y al Sistema Operativo. Se recomienda que se use la cuenta de root solo para la administración del sistema. Por estándar la contraseña debe ser fácil de recordar y al menos tener 6 caracteres alfanuméricos.

Figura 24 Establecimiento de contraseña de usuario root



The screenshot shows a text-based installation interface. At the top left is the Red Hat logo. To its right, the text reads: "La cuenta root se utiliza para la administración del sistema. Introduzca una contraseña para el usuario root." Below this, there are two input fields. The first is labeled "Contraseña de root:" and contains six asterisks. The second is labeled "Confirmar:" and also contains six asterisks, with a cursor visible at the end of the text.

Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6

3.1.3.7. Configuración del particionamiento del disco

En esta sección se puede elegir entre un sistema de particiones predeterminado entre las 4 formas disponibles o se puede realizar un particionamiento manual de acuerdo a lo que se necesite.

Las primeras 4 opciones de instalación automática permiten realizar el particionamiento del disco de forma predeterminada, sin la necesidad de preocuparse en dividir los dispositivos de almacenamiento, a menos de que se seleccione la opción "Revisar y Modificar el diseño de particiones"; con esta opción se puede modificar el particionamiento predeterminado.

➤ Usar todo el Espacio

Remueve todas las particiones en el disco duro donde se va a instalar Red Hat Enterprise Linux, esto involucra remover particiones de otros Sistemas Operativos.

➤ **Reemplazar Sistemas Linux Existentes**

Remueve únicamente las particiones Linux instaladas anteriormente

➤ **Achicar el Sistema Actual**

Reduce manualmente las particiones actuales que se encuentren en el disco, para así dar espacio a una instalación de Red Hat Enterprise Linux.

➤ **Usar Espacio Libre**

Conserva los Sistemas Operativos que se hayan instalado anteriormente y solo instala Red Hat Enterprise Linux en el espacio libre que se tenga. Se debe verificar que haya suficiente espacio libre para la instalación antes de escoger esta opción.

➤ **Crear un Diseño Personal**

Se particiona manualmente el dispositivo de almacenamiento y se crean diseños personalizados.

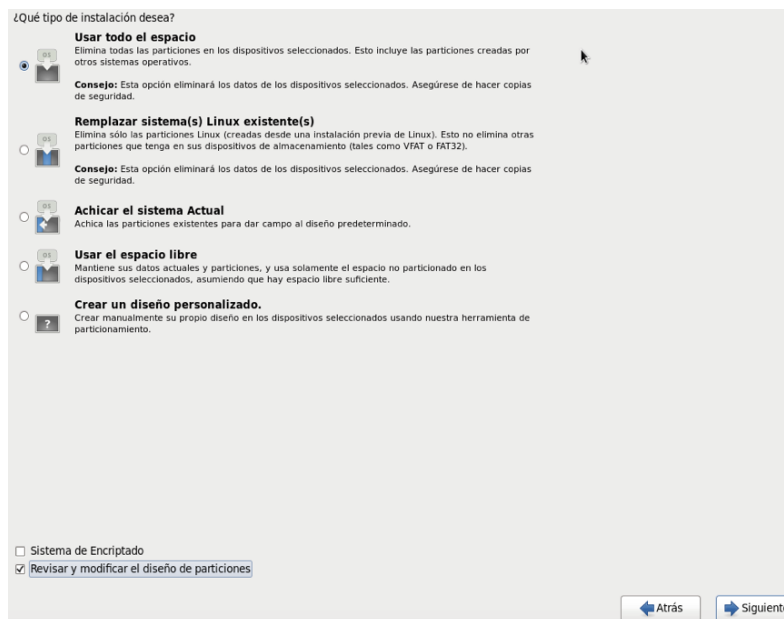
La opción “Sistema Encriptado” se utiliza para cifrar todas las particiones que se crean, con excepción del “/boot”

La opción “Revisar y Modificar el diseño de particiones” se utiliza para revisar, cambiar y modificar el diseño de las particiones que son creadas automáticamente.

Para esta guía se escogió el particionamiento “Usar todo el espacio” con la opción de Revisar y Modificar el diseño de particiones.

GUÍA METODOLÓGICA PARA LA IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA
DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES, UTILIZANDO OPEN SOURCE.
CASO DE ESTUDIO: EMPRESA SOPORTE LIBRE

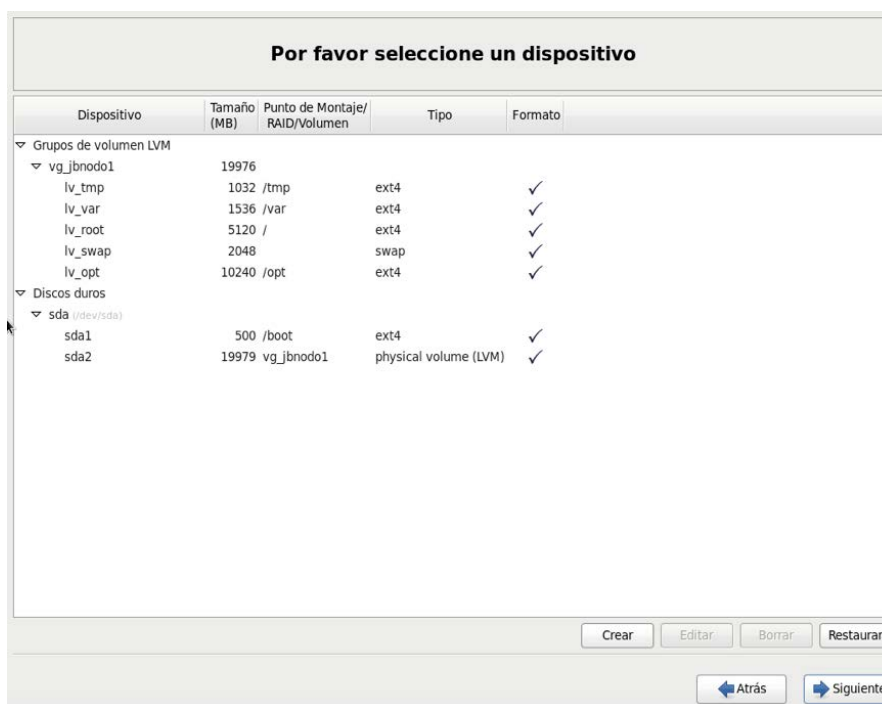
Figura 25 Configuración Particionamiento del Disco



Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6

En este caso, el particionamiento se distribuirá de acuerdo a la [tabla 3](#) y [tabla 4](#)
Nodos 1 y 2:

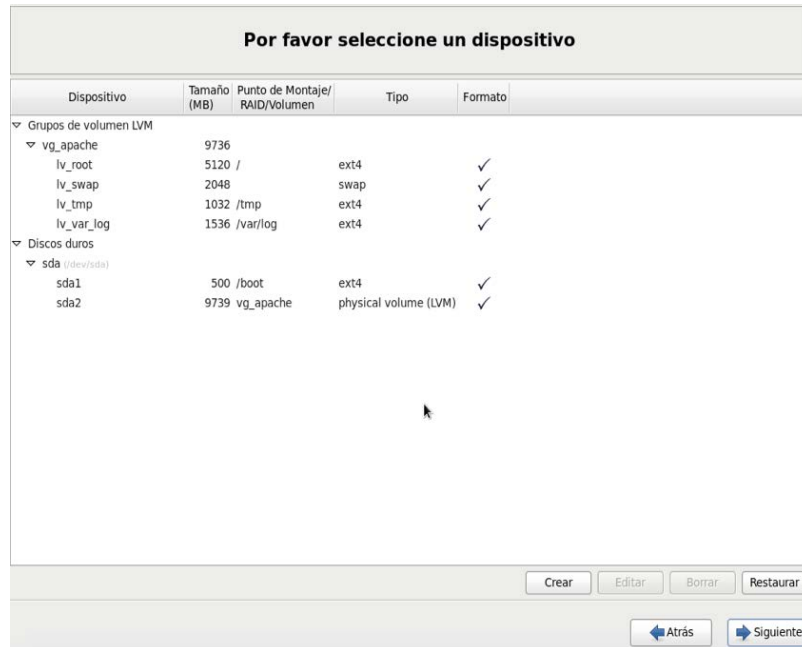
Figura 26 Configuración Particionamiento del Disco Nodo 1 y 2



Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6 para Nodo 1 y 2

Balancedor:

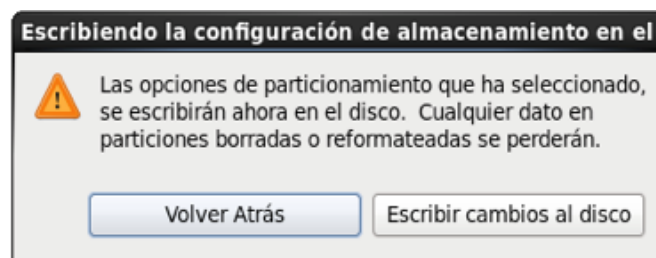
Figura 27 Configuración Particionamiento del Disco Balanceador



Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6 para Balanceador

Una vez ingresados los valores del particionamiento del sistema, se debe confirmar los cambios al disco.

Figura 28 Confirmación de Configuración Particionamiento del Disco



Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6

3.1.3.8. Configuración del gestor de arranque

Para poder el iniciar el Sistema Operativo sin los medios de arranque, se va a necesitar un gestor de arranque. Este gestor es lo primero que se ejecuta cuando se inicia el servidor; es el encargado de cargar el Sistema Operativo en memoria y transfiere el

control al kernel de Red Hat Enterprise Linux con los archivos necesarios, a continuación el kernel inicia con el resto del sistema.

En Red Hat Enterprise Linux el gestor de arranque se llama GRUB (Grand Unified Bootloader), el cual está instalado por defecto.

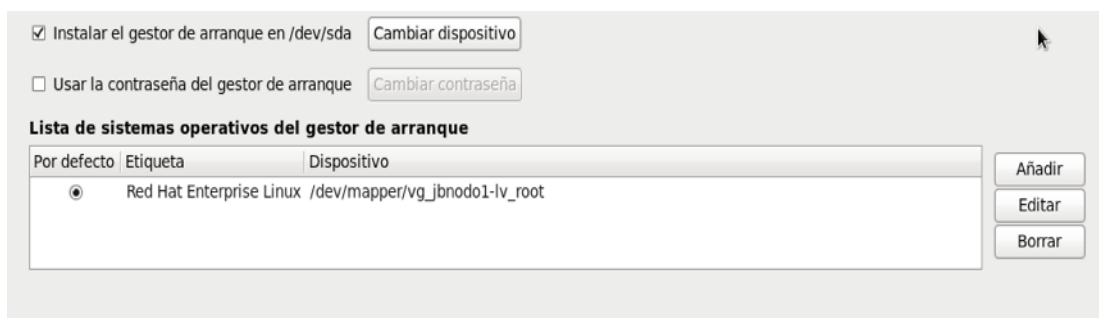
Existe la opción de establecer una contraseña al GRUB para aumentar el nivel de seguridad, para esto se selecciona la casilla “Usar la contraseña del Gestor de Arranque” y a continuación se ingresa la contraseña que se desea.

GRUB guarda la contraseña en un formato encriptado, por lo que es imposible leerla o recuperarla.

No es necesario establecer contraseña al GRUB si las personas que tienen acceso al servidor son confiables o está asegurado con acceso remoto a consola controlado.

Para esta guía, como se está en un medio controlado y de pruebas no es necesario establecer contraseña, por lo que no se selecciona la casilla.

Figura 29 Configuración del Gestor de Arranque



Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6

3.1.3.9. Selección de tipo de servidor

En el proceso de instalación de Red Hat Enterprise Linux se carga una amplia selección de software de acuerdo al uso que se le vaya a dar al servidor, las opciones son las siguientes:

➤ **Servidor Básico**

Provee herramientas básicas de Red Hat Enterprise Linux

➤ **Servidor de Base de Datos.**

Provee paquetes de MySQL y PostgreSQL, para que el servidor funcione como Base de Datos.

➤ **Servidor Web**

Esta opción viene con paquetes para que el servidor sea web Apache.

➤ **Servidor de administración de identidad**

Con esta opción el servidor viene con paquetes de OpenLDAP y SSD (Servicios de Seguridad del Sistema), esto se utiliza para que sea un servidor de autenticación de usuarios.

➤ **Host de virtualización**

En esta opción se provee herramientas de KVM y gestores de maquinas virtuales, para que este servidor sea utilizado como host de maquinas virtuales

➤ **Escritorio**

Con este tipo de instalación el servidor tiene un entorno grafico y provee programas de productividad como OpenOffice, GIMP y entre otras.

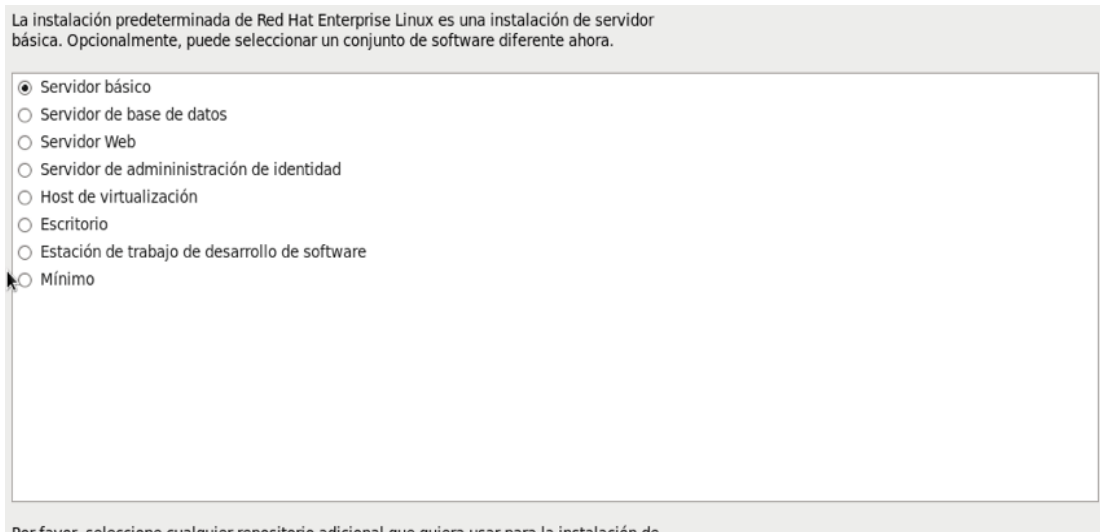
➤ **Estación de trabajo de desarrollo de software**

En esta opción el servidor tiene herramientas para desarrollo de software y compilación.

➤ **Minimal**

Este tipo de instalación provee las herramientas mínimas para que funcione Red Hat Enterprise Linux. Viene solo con paquetes esenciales.

Figura 30 Opciones de Tipo de Servidor



Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6

También se puede seleccionar repositorios adicionales, esto es para aumentar la cantidad de paquetes disponibles en el sistema Red Hat Enterprise Linux que se va instalar.

Un repositorio es un depósito en donde se almacenan y se mantienen un conjunto de paquetes. Los repositorios adiciones que se pueden seleccionar son los siguientes:

➤ **Alta Disponibilidad**

En este tipo de repositorio están todos los paquetes necesarios para que el servidor Red Hat Enterprise Linux pueda funcionar en Alta Disponibilidad.

➤ **Equilibrador de Carga**

En este repositorio se encuentran paquetes necesarios para un agrupamiento de balance de carga utilizando LVS (Linux Virtual Server).

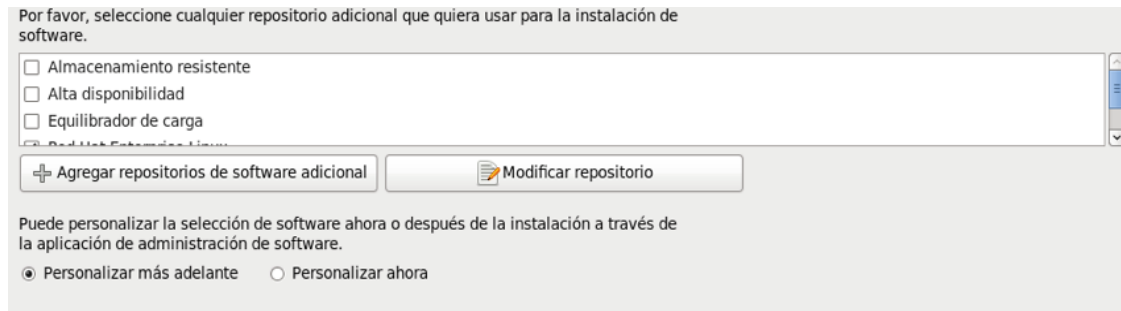
➤ **Red Hat Enterprise Linux**

Este repositorio es seleccionado automáticamente. Incluyen los paquetes de Red Hat Enterprise Linux 6, que en su momento, eran los más actuales.

➤ **Almacenamiento Resistente**

Incluye paquetes para la utilización de GFS (Sistema de Archivos Global), este se utiliza para agrupamiento de almacenaje

Figura 31 Opciones de Tipo de Servidor



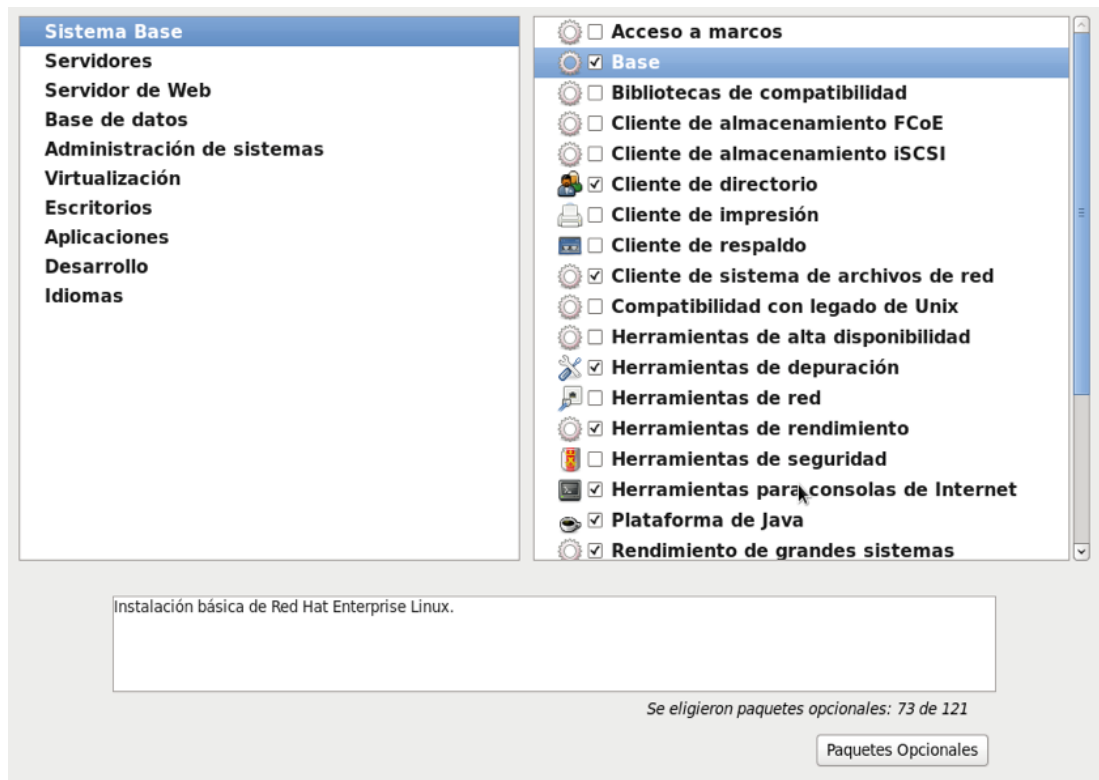
Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6

En la sección de personalizar la selección de software, se especifican paquetes adicionales que se quieran instalar en el sistema operativo, para acceder a esta sección se debe escoger la opción “**Personalizar ahora**”.

En la opción “Personalizar ahora” se tiene una pantalla extra, en la cual, para una mayor facilidad en la selección de software, se divide a los paquetes en categorías. Para instalar un grupo de paquetes que se debe marcar la casilla junto al grupo que se desea instalar. La casilla que se encuentra en el borde inferior muestra todos los detalles del grupo de paquete que se selecciono.

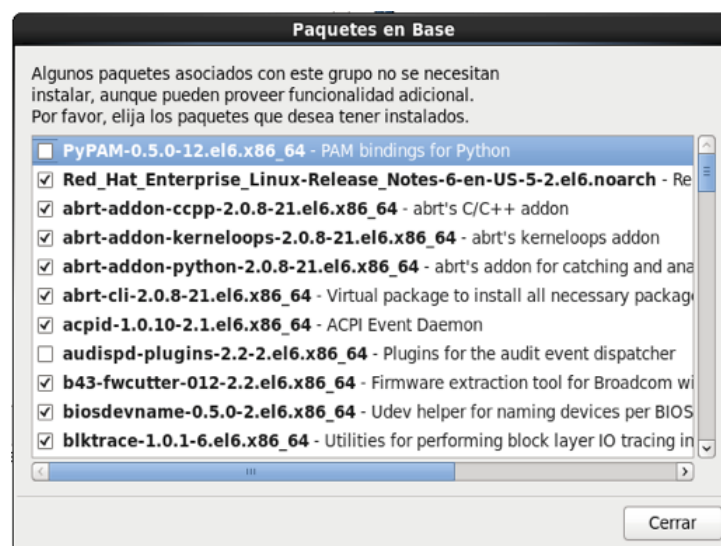
Cuando se selecciona un grupo de paquetes, automáticamente se instalan los paquetes que son imprescindibles de ese grupo. Para aumentar los paquetes que quieran instalar en dicho grupo se selecciona “**Paquetes Opcionales**”

Figura 32 Opción de Paquetes Opcionales



Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6

Figura 33 Opción de Paquetes Base



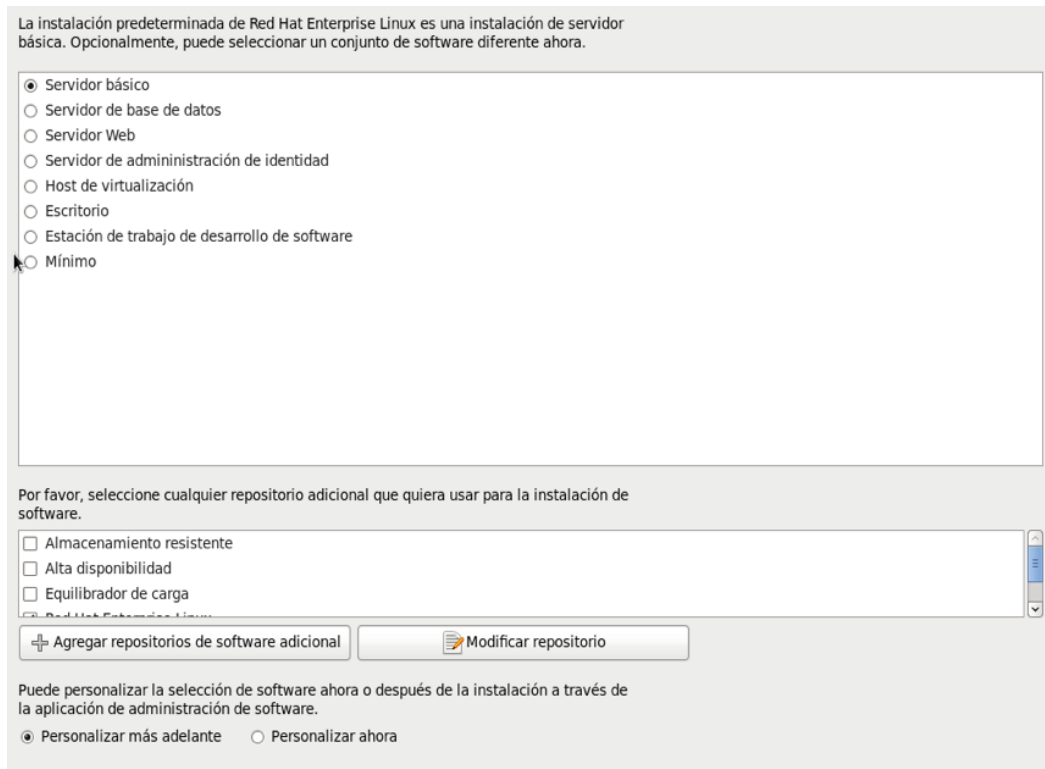
Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6

Para esta guía Metodológica se selecciona Servidor Básico, ya que no se requiere que se tenga una interfaz grafica, y los paquetes que se requieran se van a instalar

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR, FACULTAD DE 48
INGENIERÍA, ESCUELA DE SISTEMAS

posteriormente. No se selecciona ningún tipo de repositorio adicional, y se selecciona “Personalizar Después” en la sección de personalización de software.

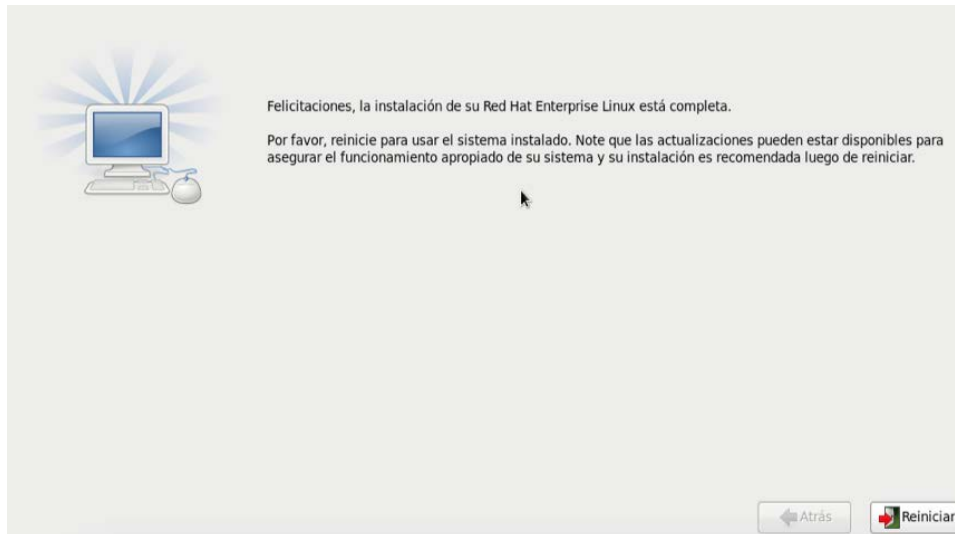
Figura 34 Selección de Tipo de Servidor



Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6

El proceso de instalación dura aproximadamente 20 minutos para terminarse. Al finalizar se tendrá un mensaje de finalización de instalación de Red Hat Enterprise Linux 6 y un mensaje de reinicio.

Figura 35 Finalización de Instalación Red Hat Enterprise Linux 6.6



Fuente: Resultado de instalación de Red Hat Enterprise Linux 6.6

3.2. Configuración de Red Hat Enterprise Linux 6.6

3.2.1. Configuración de interfaz de red

Para la configuración de red en el servidor Red Hat Enterprise Linux 6.6 se debe abrir el archivo “/etc/sysconfig/network-script/ifcfg-eth0” de la siguiente forma:

```
# vim /etc/sysconfig/network-script/ifcfg-eth0
```

Una vez en el archivo de configuración de red, se deben modificar los siguientes parámetros:

ONBOOT=yes

Este parámetro se lo activa para que cuando se inicie el servidor Red Hat Enterprise Linux 6.6, también se inicie la interfaz de red automáticamente.

NM_MANAGER=no

Se deshabilita esta opción para que el NetworkManager no tome control de la interfaz de red, esto sirve para que el usuario administrador del servidor sea quien

BOOTPROTO=static

Esto se usa para que la interfaz de red utilice una dirección IP estática, a continuación se procede a añadir los datos de IP, Gateway, Netmask, DNS

```
IPADDR=192.168.XX.XX  
NETMASK=255.255.XX.XX  
GATEWAY=192.168.XX.XX  
DNS1=192.168.XX.XX
```

Estas configuraciones de red son usadas de acuerdo a la [tabla 2](#).

Para que se apliquen los cambios realizados en el archivo de configuración se debe reiniciar el servicio de red, para ello se ejecuta el siguiente comando desde una terminal.

```
# service network restart
```

3.2.2. Configuración de resolución de nombres

Se realiza esta configuración para que el servidor sea capaz de resolver los nombres de host de servidores que se encuentran en la red interna. Para realizar esto se debe editar el archivo “**/etc/hosts**” en un terminal de la siguiente forma:

```
# vim /etc/hosts
```

En el archivo hosts se agregan todos los nombres de los nodos y del servidor Apache que van a funcionar en el clúster de Alta Disponibilidad

En este archivo se declara la IP, y el nombre de host del servidor

```
192.168.XX.XX nodo1.jboss.com nodo1  
192.168.XX.XX nodo2.jboss.com nodo2  
192.168.XX.XX balanceador.apache.com balanceador
```

3.2.3. Instalación de Paquetes

Para la instalación de los paquetes necesarios en el servidor Red Hat Enterprise Linux 6.6 se tienen dos opciones que son las siguientes:

3.2.3.1. Instalación de repositorio propio de datos

Para la configuración de un repositorio de datos propio se necesita tener el DVD de Red Hat Enterprise Linux 6.5.

GUÍA METODOLÓGICA PARA LA IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA
DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES, UTILIZANDO OPEN SOURCE.
CASO DE ESTUDIO: EMPRESA SOPORTE LIBRE

Una vez con el DVD en el servidor se procede a montar en la partición “/mnt” de la siguiente forma:

```
# mount /dev/cdrom /mnt
```

Para la configuración del repositorio de datos se procede a crear el archivo “/etc/yum.repos.d/local.repo”. El archivo contendrá la siguiente información con lo cual se tendrá acceso a los repositorios de paquetes necesarios para la instalación de JBoss EAP 6.

```
# touch /etc/yum.repos.d/local.repo  
# vim /etc/yum.repos.d/local.repo
```

```
[base]  
name=Local  
baseurl=file:///mnt/Server  
enabled=1  
gpgcheck=1  
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release  
  
[HighAvailability]  
name=HighAvailability  
baseurl=file:///mnt/HighAvailability  
enabled=1  
gpgcheck=1  
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release  
  
[LoadBalancer]  
name=LoadBalancer  
baseurl=file:///mnt/LoadBalancer  
enabled=1  
gpgcheck=1  
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

GUÍA METODOLÓGICA PARA LA IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA
DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES, UTILIZANDO OPEN SOURCE.
CASO DE ESTUDIO: EMPRESA SOPORTE LIBRE

```
[ResilientStorage]
name=ResilientStorage
baseurl=file:///mnt/ResilientStorage
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

[ScalableFileSystem]
name=ResilientStorage
baseurl=file:///mnt/ScalableFileSystem
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

Para habilitar el repositorio, y que este en funcionamiento se ejecuta en la terminal:

```
# yum clean all
# yum repolist
```

La salida debe ser parecida a lo siguiente:

```
# yum clean all
# yum repolist

id del repositorio      nombre del repositorio      estado
HighAvailability        HighAvailability            55
LoadBalancer            LoadBalancer                4
ResilientStorage        ResilientStorage            62
ScalableFileSystem      ResilientStorage            7
base                    Local                        3785
repolist: 3913
```

3.2.3.2. Registrar Servidor a Red Hat Network¹

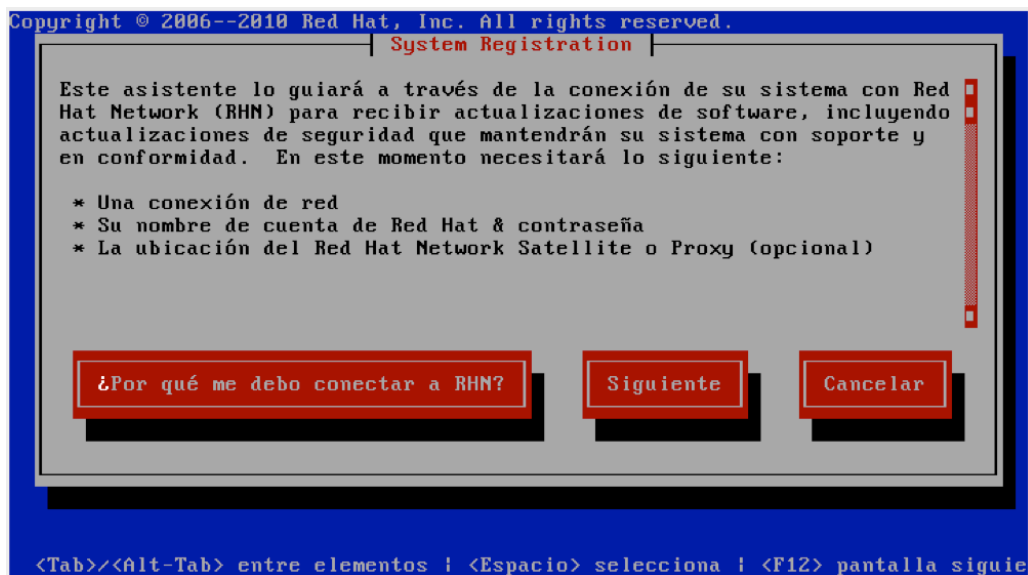
Una de las opciones con las que se puede tener acceso a nuevos paquetes es suscribiendo el servidor a Red Hat Network. Pero para tener acceso a esto, se requiere de suscripciones, que varían de acuerdo a las características propias de cada servidor. La ventaja de estar suscrito a Red Hat Network es recibir actualizaciones y parches de seguridad.

En una terminal se ejecuta el siguiente comando:

```
# rhn_register
```

Se desplegará la siguiente pantalla del sistema registro del equipo a Red Hat Network

Figura 36 Inicio de Registro a Red Hat Network



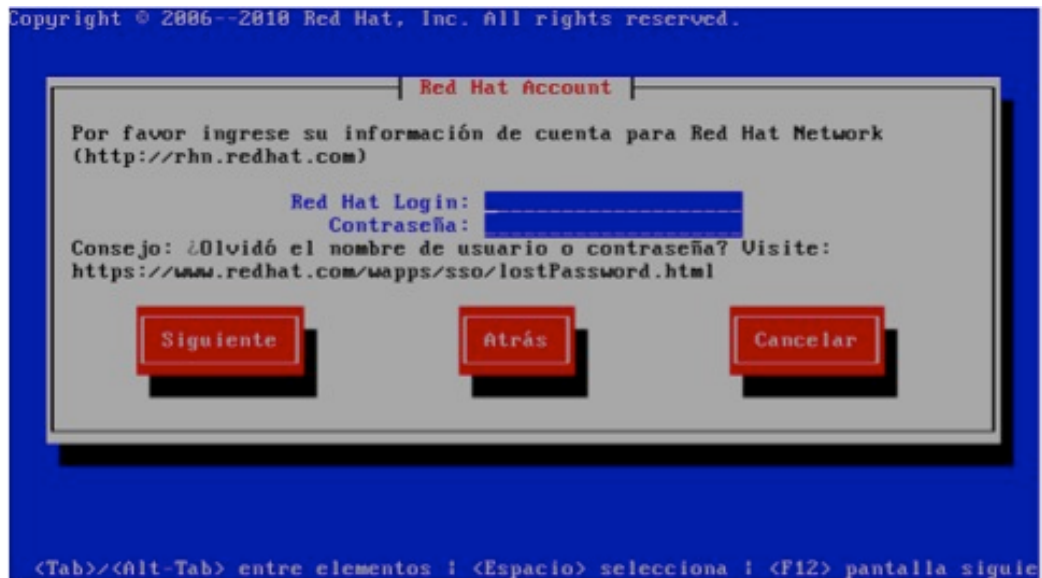
Fuente: Soporte Libre Cía. Ltda.

Realizado: Gabriel Andrade

¹ Las figuras 36, 37, 28, 29, 40, 41, 42, 43 y 44 que muestran el procedimiento para registrar un servidor a Red Hat Network se obtuvieron utilizando la credencial de Soporte Libre Cía. Ltda. Con el debido permiso. Estas imágenes pertenecen al manual de Registro de Servidor a Red Hat Network creado por Gabriel Andrade

Se ingresa el nombre de usuario y contraseña

Figura 37 Ingreso de usuario y contraseña

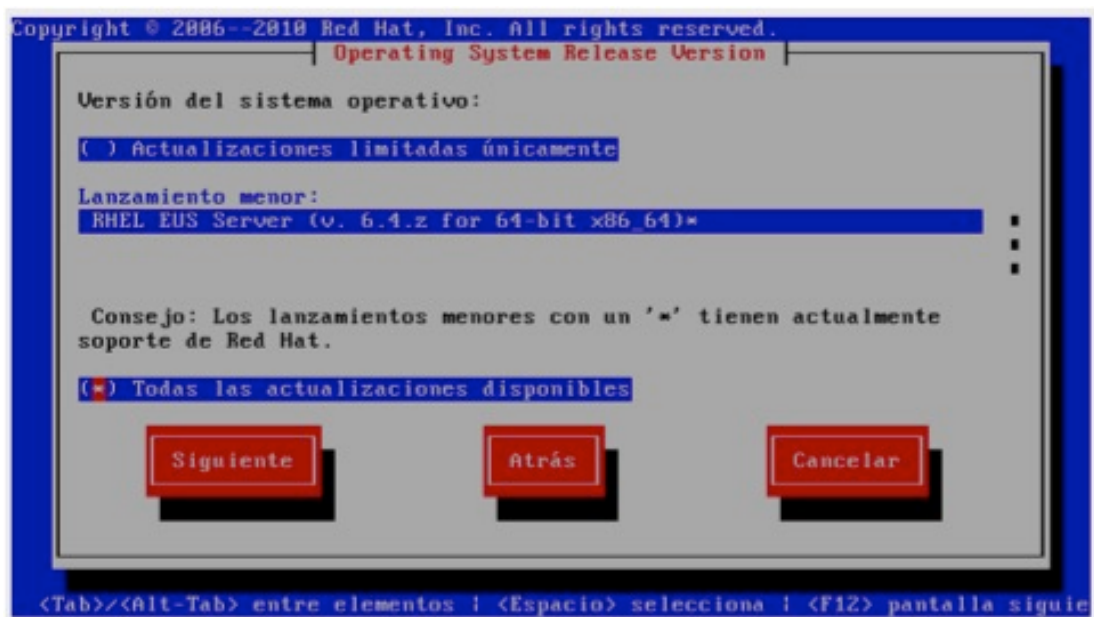


Fuente: Soporte Libre Cía. Ltda.

Realizado: Gabriel Andrade

Se selecciona la versión del sistema operativo.

Figura 38 Selección de Versión de Sistema Operativo

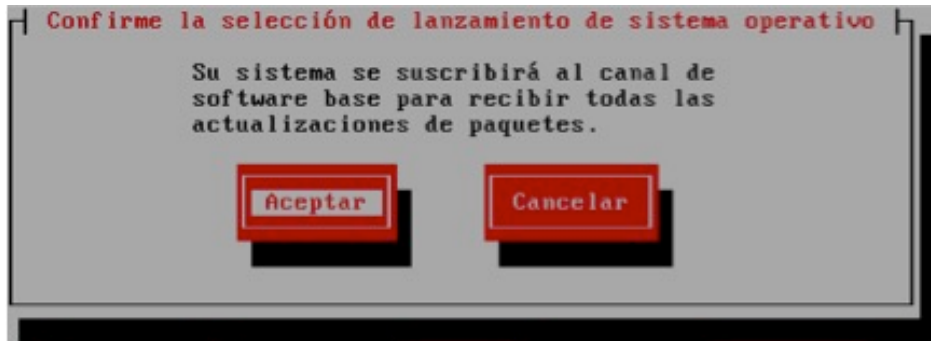


Fuente: Soporte Libre Cía. Ltda.

Realizado: Gabriel Andrade

Se confirma las actualizaciones que recibirá el servidor.

Figura 39 Confirmación de actualizaciones



Fuente: Soporte Libre Cía. Ltda.

Realizado: Gabriel Andrade

Se procede a dar un nombre del perfil del servidor con el que será identificado en el RedHat Network.

Figura 40 Nombre del perfil de Servidor

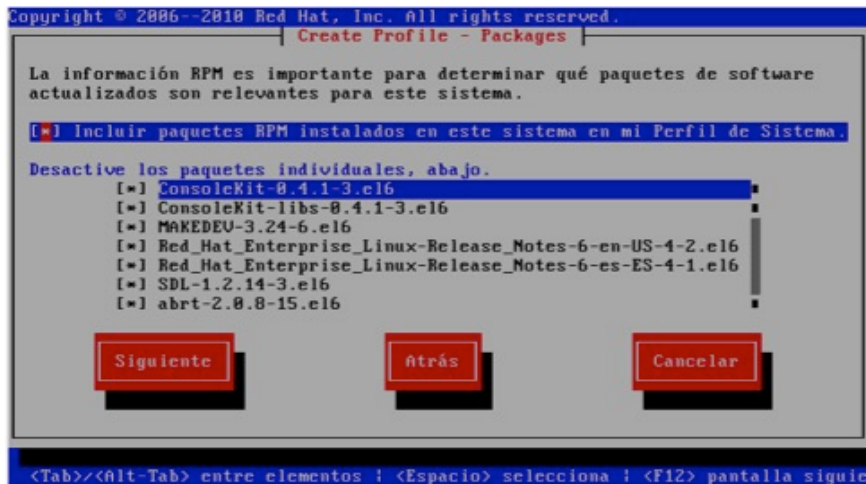


Fuente: Soporte Libre Cía. Ltda.

Realizado: Gabriel Andrade

Por seguridad se acepta que Red Hat Network tenga un registro de paquetes instalados en el servidor

Figura 41 Confirmación de registro de paquetes instalados en servidor



Fuente: Soporte Libre Cía. Ltda.

Realizado: Gabriel Andrade

Se envía la Información del perfil del servidor a Red Hat Network

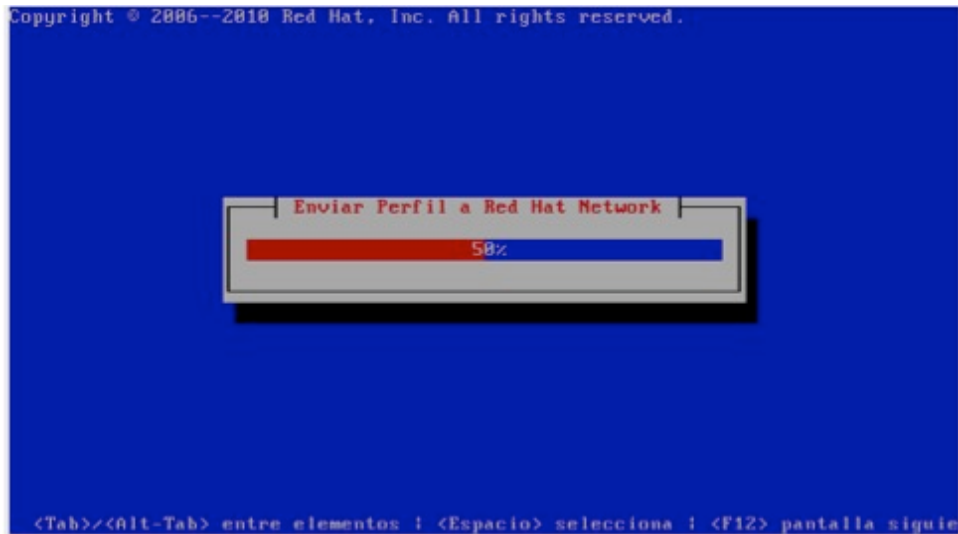
Figura 42 Finalización de registro a Red Hat Network



Fuente: Soporte Libre Cía. Ltda.

Realizado: Gabriel Andrade

Figura 43 Envío de información a Red Hat Network

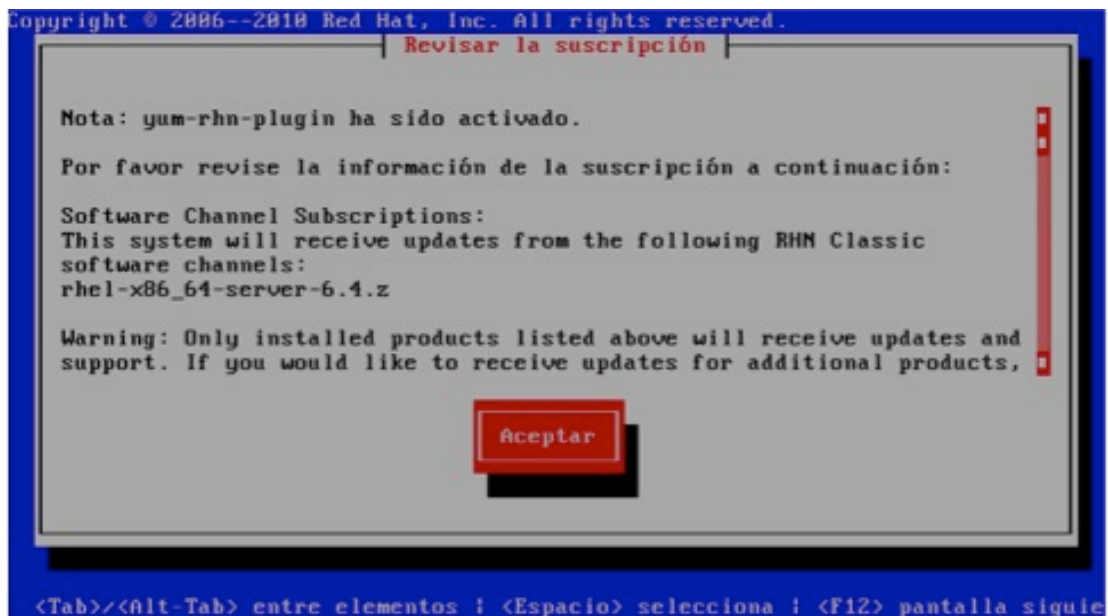


Fuente: Soporte Libre Cía. Ltda.

Realizado: Gabriel Andrade

Finalmente se presenta un resumen informativo del registro del servidor a RedHat Network

Figura 44 Confirmación de suscripción a Red Hat Network



Fuente: Soporte Libre Cía. Ltda.

Realizado: Gabriel Andrade

3.3. Configuración de Servidor de Balanceo

El servidor de balanceo nos permite hacer una re-distribución de las peticiones realizadas a los servidores de aplicaciones permitiendo mejorar el rendimiento de las aplicaciones publicadas a través de esta capa de balanceo de carga.

3.3.1. Instalación de paquetes necesarios para la configuración del servidor de balanceo

Para el servidor de Balanceo se necesita dos repositorios adicionales al repositorio previamente se configurado. Se utiliza el repositorio de EPEL y RPM Fusion; estos dos repositorios son Open Source basados en el proyecto comunitario de Fedora, proveen paquetes para distribuciones de Linux como Red Hat Enterprise Linux.

Para añadir estos repositorios de debe ejecutar los siguientes comandos en una terminal

```
# cd /tmp
# wget http://download.fedoraproject.org/pub/epel/6/x86\_64/epel-release-6-8.noarch.rpm
#yum localinstall epel-release-6-8.noarch.rpm
#yum localinstall --nogpgcheck
http://download1.rpmfusion.org/free/el/updates/6/i386/rpmfusion-free-release-6-1.noarch.rpm
http://download1.rpmfusion.org/nonfree/el/updates/6/i386/rpmfusion-nonfree-release-6-1.noarch.rpm
#yum clean all
#yum repolist
```

Se procede a instalar los paquetes necesarios

```
# yum install httpd modcluster mod_ssl mod_dnssd mod_dav_svn cluster-snmpp mod_perl mod_cluster
```

3.3.2. Configuración del servidor de Balanceo de Carga

Los archivos de configuración para el balanceo de carga son los siguientes:

3.3.2.1. httpd.conf

En el archivo de configuración del Apache se debe deshabilitar un modulo para que no haya incompatibilidad en un modulo que va a ser usado en el archivo de configuración del mod_cluster.

```
# vim /etc/httpd/conf/httpd.conf  
  
#LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
```

3.3.2.2. mod_cluster.conf

Se debe crear el archivo “mod_cluster.conf” en el directorio httpd; el archivo contendrá las configuraciones del balanceador. Incluye las directivas necesarias para las aplicaciones que van a ser balanceadas. El servidor web es el encargado de frontear las peticiones del usuario a las aplicaciones que se encuentran ejecutando en el servidor de aplicaciones JBoss EAP 6.3

El archivo mod_cluster.conf contiene lo siguiente:

En la primera sección se declara los módulos que van a ser utilizados para el funcionamiento del balanceador.

Se crea un Virtual Host el cual escucha una IP y un puerto específico, para esta guía se utiliza la IP del balanceador que es la del Apache y el puerto por defecto, que es el 80.

- **Allow from all:** Permite conexiones entrantes desde cualquier IP.
- **ManagerBalancerName:** Es el nombre del balanceador, con el cual será reconocido por los nodos JBoss que quieran conectarse.
- **KeepAliveTimeout:** Son los segundos que el Apache esperara para una petición posterior antes de cerrar la conexión. Una vez que se realice una petición, se esperara los segundos especificados para una próxima petición antes de cerrar la conexión.

- **MaxKeepAliveRequests:** Es el numero de peticiones permitidas por cada conexión cuando el parámetro KeepAlive esta activado. Se recomienda que este parámetro sea un valor alto para obtener un mejor rendimiento en el servidor.
- **EnableMCPMReceive:** Permite al VirtualHost recibir MCPM (protcolo de mensajes mod_cluster) de los nodos que se encuentran conectados al clúster. Se necesita Habilitar este parámetro para que mod cluster pueda funcionar correctamente.
- **ServerAdvertise:** Se utiliza para enviar la información del clúster. Se especifica la IP del servidor que recibe la información y por que puerto. Por defecto el estado es Off.
- **LogLevel:** Se especifica a partir de que nivel de error será grabado los logs. Existen los siguientes niveles de error, en orden de importancia:

Tabla 5 Niveles de notificación mod_cluster

Nivel	Descripción
emerg	Condiciones de emergencias, el sistema se encuentra fuera de uso
alert	Se deben tomar medidas de inmediato
crit	Condiciones criticas
error	Condiciones de error, inidican una interrupción en una solicitud o la cacacidad de atender un servicio
warn	Condiciones de alertas, indican un error de servicio no crítico
notice	Condiciones normales pero significativas
info	Condiciones informativas
debug	Registra todos los eventos que sucedan

- **Definición de Contextos:** Para definir los contextos que se van a acceder en el URL se debe declarar "<Location /contexto>". En esta sección se declaran todos los permisos que se tendrá para acceder. El parámetro "Allow from" se describe

quienes van a poder acceder, se puede usar “all” el cual permite que cualquier IP pueda acceder al contexto o se describe IPs específicas que pueden acceder.

- **SetHandler mod_cluster-manager:** Se utiliza para desplegar la información acerca de los nodos JBoss que están conectados al Balanceador y forman el clúster, adicionalmente realiza un conteo del numero de sesiones activas.

```
# vim /etc/httpd/conf.d/mod_cluster.conf

LoadModule slotmem_module modules/mod_slotmem.so
LoadModule manager_module modules/mod_manager.so
LoadModule proxy_cluster_module modules/mod_proxy_cluster.so
LoadModule advertise_module modules/mod_advertise.so

<VirtualHost 10.0.1.46:80>
ManagerBalancerName cluster_tesis

    KeepAliveTimeout 300
    MaxKeepAliveRequests 0
    EnableMCPMReceive On
    ServerAdvertise Off
    LogLevel debug

    <Location /mod_cluster-manager>
        SetHandler mod_cluster-manager
        Order deny,allow
        Allow from all
    </Location>
</VirtualHost>
```

3.3.3. Configuración de Iptables

Se debe configurar las reglas del iptables del servidor, para que permita la conexión con los nodos JBoss; se debe habilitar los puertos 80, 8080, 443.

En Red Hat Enterprise Linux 6.6 se debe editar el archivo de configuración de las iptables en donde se encuentran todas las reglas que están funcionando y se debe añadir los siguientes comandos para habilitar los puertos:

```
# vim /etc/sysconfig/iptables
-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8080 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
```

Se necesita reiniciar el servicio iptables para que las nuevas reglas entren en funcionamiento.

```
#service iptables restart
iptables: Poniendo las cadenas de la política ACCEPT: filte[ OK ]
iptables: Guardando las reglas del cortafuegos: [ OK ]
iptables: Descargando módulos: [ OK ]
iptables: Aplicando reglas del cortafuegos: [ OK ]
# service iptables save
iptables: Guardando las reglas del cortafuegos en /etc/sysc[ OK ]tables:
```

3.3.4. Inicialización del Apache

Para iniciar el servicio de Apache se debe ejecutar el siguiente comando:

```
# service httpd start
```

Para que el servicio del Apache sea iniciado con el sistema operativo Red Hat Enterprise Linux 6.6 se ejecuta el siguiente comando:

```
# chkconfig httpd on
```

3.4. Instalación de JBoss EAP 6.3

Para un correcto funcionamiento del servidor de aplicaciones JBoss EAP 6.3 se necesita realizar instalaciones y configuraciones previas que se detallan a continuación:

3.4.1. Requisitos para funcionamiento de Servidor de aplicaciones

3.4.1.1. Instalación de OpenJDK

Para poder ejecutar la plataforma JBoss EAP 6 requiere Java 6 o Java 7, con el kit de desarrollo java (JDK) y entorno de ejecución Java (JRE).OpenJDK es una de las varias alternativas de kits de desarrollo java (JDKs) que son soportadas por Red Hat Enterprise Linux.

En esta guía se va a utilizar la versión de Java OpenJDK 7, debido a que es la versión más actual y tiene una mayor compatibilidad con aplicaciones modernas.

Para realizar la instalación se debe ejecutar desde una terminal el siguiente comando con el usuario root:

```
# yum install -y java-1.7.0-openjdk.x86_64 java-1.7.0-openjdk-devel.x86_64
```

A continuación se debe establecer la variable de entorno JAVA_HOME; para ello primero se debe editar el archivo “ java.conf ” y cambiar los siguientes parámetros

```
# vim /etc/java/java.conf  
JAVA_HOME=$JVM_ROOT/jre
```

Se crea el archivo “ /etc/profile.d/java.sh ” con el siguiente contenido:

Esto se lo realiza para que JAVA_HOME este disponible para todos los usuarios registrados en el sistema.

```
# vim /etc/profile.d/java.sh  
. /etc/java/java.conf  
export JAVA_HOME
```

Se debe refrescar la variable de entorno configurada para que ya este disponible

```
# source /etc/profile.d/java.sh
```

Se debe cambiar la versión de Java predeterminada para las aplicaciones, para esto Red Hat Enterprise Linux tiene una herramienta llamada “alternatives”, para usarla se debe ejecutar los siguientes comandos:

```
# alternatives --config java
```

Con esto se desplegará una lista con todas las versiones de java que se encuentran instaladas, a continuación se procede a seleccionar la que se va a utilizar por defecto “/usr/lib/jvm/jre-1.7.0-openjdk.x86_64/bin/java”

3.4.2. Instalación de Servidor de Aplicaciones JBoss EAP 6.3

3.4.2.1. Utilización de archivo comprimido (zip) de JBoss EAP 6.3

Se debe descargar el archivo comprimido zip desde el portal oficial de servicio al cliente de Red Hat.

A continuación se procede a mover el archivo comprimido zip al servidor, en el directorio en donde se alojara la aplicación JBoss 6. Se debe tener en cuenta que este directorio debe ser accesible para el usuario que va a iniciar y detener la aplicación.

Se va a crear el usuario “jboss” que va a ser el encargado del manejo de JBoss 6

```
# useradd jboss
# passwd jboss
Changing password for user jboss.
New password: xxxxxxxx
Retype new password: xxxxxxxx
passwd: all authentication tokens updated successfully.
```

Para esta guía se va a utilizar el directorio “/opt” para que se aloje la aplicación JBoss EAP 6.3. Se procede a mover el archivo comprimido zip al directorio escogido y a continuación se descomprime.

Ejecutar en la terminal lo siguiente

```
# mv jboss-eap-6.3.0.zip /opt/
# cd /opt
# unzip jboss-eap-6.3.0.zip
```

```
.....  
inflating: jboss-eap-6.3/welcome-content/documentation.html  
inflating: jboss-eap-6.3/welcome-content/index_noconsole.html  
inflating: jboss-eap-6.3/welcome-content/index.html  
inflating: jboss-eap-6.3/welcome-content/noconsole.html  
inflating: jboss-eap-6.3/LICENSE.txt  
inflating: jboss-eap-6.3/JBossEULA.txt  
inflating: jboss-eap-6.3/jboss-modules.jar  
# ls  
jboss-eap-6.3 jboss-eap-6.3.0.zip
```

Se debe cambiar los permisos de la carpeta “jboss-eap-6.3” para el usuario “jboss”

```
# chown -R jboss:jboss jboss-eap-6.3  
# ls -ltr  
drwxrwxr-x. 10 jboss jboss    4096 May 29  2012 jboss-eap-6.3  
-rw-r-----. 1 root  root   114875787 Jan 18 18:41 jboss-eap-6.3.0.zip
```

El directorio que fue extraído “jboss-eap-6.3” se le conoce con el nombre de **EAP_HOME**. Si alguna vez se requiere utilizar la aplicación JBoss 6 en otro servidor solo se debe mover el directorio “jboss-eap-6.0”.

3.5. Estructura de JBoss EAP 6.3

En el directorio de JBoss EAP 6.3 se encuentra de forma simplificada y organizada los archivos de configuración para que sea fácil para el usuario realizar modificaciones de acuerdo al uso que se le quiera dar. A continuación se detalla los subdirectorios que están dentro de JBoss:

- **appcient:** Se utiliza para comunicaciones con un Application Server usando RMI/IIOP
- **bin:** Se encuentran todos los scripts para inicializar el servidor de aplicaciones JBoss 6.
- **bundles:** Se encuentran todos OSGi, que forman parte de las funcionalidades internas del servidor de aplicaciones JBoss 6.3.

- **docs:** Se encuentran archivos con licencias y ejemplos de configuraciones.
- **domain:** Contiene los archivos de configuración, el contenido de la implementación que se quiere desplegar cuando se desea utilizar JBoss 6 en modo domain
- **modules:** Contiene todos los módulos que se necesitan para que la aplicación sea desplegada

3.6. Utilización de JBoss EAP 6.3 como servicio

Es recomendable que el servidor de aplicaciones JBoss EAP 6.3 sea ejecutado como un servicio, esto se lo realiza para que la instancia del servidor se mantenga en funcionamiento aun cuando se termine la sesión con el usuario y para que el servidor de aplicaciones JBoss EAP 6.3 se inicie automáticamente cuando el servidor sea encendido.

Existe un script en el directorio del servidor de aplicaciones JBoss, el cual debe ser modificado de acuerdo a los requerimientos que se deseen.

Lo primero que se debe hacer es ingresar al directorio de los scripts

```
# cd /opt/jboss-eap-6.3/bin/init.d/  
# ls  
jboss-as-standalone.sh jboss-as.conf jboss-as-domain.sh
```

En el directorio se encuentra el script de arranque del servidor de aplicaciones JBoss y un archivo de configuración

Se debe realizar una copia del script en el directorio “/etc/init.d” con el nombre de “jboss-eap”, se realiza el cambio de nombre para una fácil utilización.

```
# cp jboss-as-standalone.sh /etc/init.d/jboss-eap
```

Se debe agregar el nuevo servicio “jboss-eap” a los servicios que son iniciados con el sistema operativo Red Hat Enterprise Linux 6.6.

```
# cd /etc/init.d/  
# chkconfig --add jboss-eap
```

```
# chkconfig jboss-eap on
```

Se debe crear un nuevo directorio en “/etc” para el archivo de configuración inicial del servidor de JBoss y copiar el archivo “jboss-as.conf” a este nuevo directorio.

```
# cd /opt/jboss-eap-6.3/bin/init.d/  
# mkdir /etc/jboss-as  
# cp jboss-as.conf /etc/jboss-as/  
# cd /etc/jboss-as/
```

El archivo “jboss-as.conf” se utiliza para personalizar las opciones de arranque del servidor de aplicaciones JBoss. Contiene comentarios como guía para una fácil personalización.

JBOSS_USER: El usuario que se utilizara para levantar el servicio del servidor de aplicaciones JBoss.

JBOSS_HOME: La ubicación en donde se encuentra el servidor de aplicaciones JBoss.

JBOSS_CONFIG: Es el archivo con la configuración que se utilizara para el servidor de aplicaciones JBoss.

JBOSS_SCRIPT: El script que se utiliza para iniciar el servidor de aplicaciones JBoss.

Para esta guía se utilizaran los siguientes valores:

```
# vim jboss-as.conf  
JBOSS_USER=jboss  
JBOSS_HOME= /opt/jboss-eap-6.3  
JBOSS_CONFIG=standalone-ha.xml
```

3.7. Arranque de JBoss EAP 6.3

Para la inicialización del servidor de aplicaciones JBoss EAP 6.3 se debe seguir los siguientes pasos:

3.7.1. Configuración de Iptables

Se debe configurar las reglas del iptables, para que permita la conexión con los otros nodos JBoss y se pueda conectar con el servidor balanceador; se debe habilitar los puertos 80,443, 7600 y la IP de los otros nodos JBoss

En Red Hat Enterprise Linux 6.6 se debe editar el archivo de configuración de las iptables en donde se encuentran todas las reglas que están funcionando y se debe añadir los siguientes comandos para habilitar los puertos y la IP:

```
# vim /etc/sysconfig/iptables
-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 7600 -j ACCEPT
-A INPUT -s 192.168.XX.XX -j ACCEPT
```

Se necesita reiniciar el servicio iptables para que las nuevas reglas entren en funcionamiento.

```
#service iptables restart
iptables: Poniendo las cadenas de la política ACCEPT: filte[ OK ]
iptables: Guardando las reglas del cortafuegos: [ OK ]
iptables: Descargando módulos: [ OK ]
iptables: Aplicando reglas del cortafuegos: [ OK ]
# service iptables save
iptables: Guardando las reglas del cortafuegos en /etc/sysc[ OK ]tables:
```

3.7.2. Creación de Usuario inicial para consola de Administración

En la consola de administración del JBoss EAP 6.3 no existe un usuario predeterminado, es altamente recomendable crear usuarios administradores por temas de seguridad para la gestión de las aplicaciones. Se debe seguir los siguientes pasos:

En una terminal se debe ejecutar lo siguiente:

Se debe ingresar al directorio “/bin” del JBoss

```
# cd /opt/jboss-eap-6.3/bin
```

GUÍA METODOLÓGICA PARA LA IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA
DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES, UTILIZANDO OPEN SOURCE.
CASO DE ESTUDIO: EMPRESA SOPORTE LIBRE

Se ejecuta el script para la creación de usuario

```
# ./add-user.sh
```

Se presentan dos opciones para la creación de usuarios “Management User” y “Application User”; los usuarios creados en “Management” son usados para tareas de administración en el servidor de aplicaciones JBoss Enterprise Platform y se agregan **ManagementRealm**, mientras que los usuarios creados en “Application” permiten roles particulares para la aplicación que va a trabajar dentro del servidor de aplicaciones JBoss y se agregan al **ApplicationRealm**.

En esta guía se va a crear usuario de administración de JBoss, por lo tanto se escoge la opción “a”

```
What type of user do you wish to add?  
a) Management User (mgmt-users.properties)  
b) Application User (application-users.properties)  
(a): a
```

La siguiente sección se escoge el dominio al cual va a pertenecer el usuario que se está creando. Para que un usuario tenga permisos administrativos en el servidor de aplicaciones JBoss se selecciona la opción predeterminada, **ManagementRealm**.

Como en esta guía es el único dominio que está creado, automáticamente el sistema escoge **ManagementRealm**.

```
Enter the details of the new user to add.  
Using realm 'ManagementRealm' as discovered from the existing property  
files.
```

Se ingresa el nombre del nuevo usuario y la contraseña. Es recomendable por seguridad utilizar una contraseña con una combinación entre números y letras. A continuación se escoge el grupo al cual va a pertenecer el usuario que se está creando o se puede dejar en blanco lo que indica que no pertenece a ningún grupo.

También se pide una confirmación para ingresar al usuario al dominio ManagementRealm, lo cual se acepta y se escoge **yes**.

Username : usuario

Password requirements are listed below. To modify these restrictions edit the
add-user.properties configuration file.

- The password must not be one of the following restricted values {root,
admin, administrator}
- The password must contain at least 8 characters, 1 alphabetic character(s),
1 digit(s), 1 non-alphanumeric symbol(s)
- The password must be different from the username

Password : xxxxx

Re-enter Password : xxxxx

What groups do you want this user to belong to? (Please enter a comma
separated list, or leave blank for none)[]:

About to add user ' usuario ' for realm 'ManagementRealm'

Is this correct yes/no? Yes

Added user 'usuario' to file '/opt/jboss-eap-

6.3/standalone/configuration/mgmt-users.properties'

Added user 'usuario' to file '/opt/jboss-eap-6.3/domain/configuration/mgmt-
users.properties'

Added user 'usuario' with groups to file '/opt/jboss-eap-
6.3/standalone/configuration/mgmt-groups.properties'

Added user 'usuario' with groups to file '/opt/jboss-eap-
6.3/domain/configuration/mgmt-groups.properties'

Finalmente se debe ingresar si el usuario creado se va a utilizar para conectarse a otro
proceso de un JBoss, como por ejemplo como un host controller.

Is this new user going to be used for one AS process to connect to another
AS process?

e.g. for a slave host controller connecting to the master or for a Remoting
connection for server to server EJB calls.

yes/no?

3.7.3. Inicialización de JBoss EAP 6.3

3.7.3.1. Mecanismo de arranque en Standalone Mode

Para iniciar el servicio del servidor de aplicaciones JBoss 6.3, s solo se debe ejecutar el siguiente comando en una terminal del nodo de JBoss.

```
# service jboss-eap start
```

3.8. Prueba de Funcionamiento JBoss EAP 6.3

Para la comprobación de que el servidor de aplicaciones esta funcionando correctamente se debe ejecutar el siguiente comando en una terminal del nodo de JBoss:

```
# ps aux | grep java
jboss      1750 15.4 10.7 2440428 422556 ?        Ss   21:31   0:17
/usr/lib/jvm/jre/bin/java -D[Standalone] -server -XX:+UseCompressedOops -
verbose:gc          -Xloggc:/opt/jboss-eap-6.3/standalone/log/gc.log -
XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+UseGCLogFileRotation
-XX:NumberOfGCLogFiles=5          -XX:GCLogFileSize=3M          -XX:-
TraceClassUnloading -Xms1303m -Xmx1303m -XX:MaxPermSize=256m -
Djava.net.preferIPv4Stack=true          -
Djboss.modules.system.pkgs=org.jboss.byteman -Djava.awt.headless=true -
Djboss.modules.policy-permissions=true -Dorg.jboss.boot.log.file=/opt/jboss-
eap-6.3/standalone/log/server.log -Dlogging.configuration=file:/opt/jboss-eap-
6.3/standalone/configuration/logging.properties -jar /opt/jboss-eap-6.3/jboss-
modules.jar -mp /opt/jboss-eap-6.3/modules -jaxpmodule javax.xml.jaxp-
provider org.jboss.as.standalone -Djboss.home.dir=/opt/jboss-eap-6.3 -
Djboss.server.base.dir=/opt/jboss-eap-6.3/standalone -c standalone-ha.xml
root      1943  0.0  0.0 103216   812 pts/0    S+   21:33   0:00 grep java
```

Con este comando se obtiene un status de la instancia levantada en el servidor de aplicaciones JBoss EAP 6.3; se obtiene información como el usuario con el que esta levantado la instancia, con que archivo de configuración esta funcionando, el directorio del log y los módulos que están activos.

Para una verificación de que no existan errores en el levantamiento del servidor de aplicaciones JBoss, se debe revisar el log.

3.9. Configuración de Clúster en Standalone Mode

Como se menciona en el Capítulo 1, el servidor de Aplicaciones JBoss EAP 6.3 centraliza todas sus configuraciones en un único archivo. Para la configuración de clústeres se va a utilizar el archivo de “standalone-ha.xml”, ya que viene integrado los subsistemas que se necesitan para alta disponibilidad. Este archivo se encuentra en el directorio de las configuraciones de standalone del JBoss.

Las configuraciones que se necesitan realizar son las siguientes:

- Se debe editar el archivo standalone-ha.xml:

```
# vim /opt/jboss/standalone/standalone-ha.xml
```

- Cada nodo debe tener un identificador dentro del clúster, el cual se lo configura así:

```
<server name="standalone-nodoX" xmlns="urn:jboss:domain:1.6">
```

Donde “X” es el número de nodo.

El atributo **xmlns** especifica el nombre XML que va a ser utilizado por el subsistema de JGroups y la versión.

- Se debe crear un stack de protocolos, en donde se definen los nodos que forman parte del clúster con la siguiente configuración:

```
<stack name="true-tcp">
  <transport type="TCP" socket-binding="jgroups-tcp"/>
  <protocol type="TCPPING">
    <property name="initial_hosts">
      192.168.9.61[7600],192.168.9.62[7600]
    </property>
  </protocol>
  <protocol type="MERGE2"/>
  <protocol type="FD SOCK" socket-binding="jgroups-tcp-fd"/>
  <protocol type="FD"/>
  <protocol type="VERIFY_SUSPECT"/>
```

```
<protocol type="pbcast.NAKACK"/>  
<protocol type="UNICAST2"/>  
<protocol type="pbcast.STABLE"/>  
<protocol type="pbcast.GMS"/>  
<protocol type="UFC"/>  
<protocol type="MFC"/>  
<protocol type="FRAG2"/>  
</stack>
```

stack name: Es el nombre del grupo de protocolos a utilizarse para la comunicación entre los nodos.

Transport type: El tipo comunicación que se va a usar para el envío de mensajes entre los miembros del clúster.

Socket-binding: Se define el socket por donde el servidor va a aceptar las conexiones entrantes para la comunicación entre los nodos. Por defecto el puerto que utiliza es el 7600.

TCPPING: Transporta por TCP; utiliza una lista definida de miembros que forman el clúster.

En el parámetro "initial host" se declaran las IPs y los puertos utilizados por el socket-binding definido anteriormente.

MERGE2: Este protocolo se utiliza cuando por alguna razón el clúster es separado, como por ejemplo una partición en la red. MERGE2 se encarga de volver a unir a los sub-clústeres en uno solo. (Red Hat, Inc., 2013)

FD: Se utiliza para Failure Detection (detección de errores) por medio de mensajes entre los nodos del clúster. (Red Hat, Inc., 2013)

FD SOCK: Está basado en anillos de sockets TCP creados entre los miembros del clúster. Los nodos se conectan entre sí y forman un anillo. Para el envío de mensajes y detección de errores utilizan el socket-binding "jgroups-tcp-fd" que utiliza el puerto 57600. (Red Hat, Inc., 2013)

VERIFY_SUSPECT: Verifica que un nodo del clúster esté realmente inactivo mediante envío de paquetes ICMP (ping) antes de excluirlo del clúster. (Red Hat, Inc., 2013)

pbcast.NAKACK: Provee las propiedades para envío de mensajes entre los nodos mediante el método FIFO (first in first out), para garantizar la entrega de mensajes. (Red Hat, Inc., 2013)

UNICAST2: Garantiza la entrega de paquetes sin pérdida de información en la comunicación entre dos nodos del clústeres. (Red Hat, Inc., 2013)

pbcast.STABLE: Se encarga de recolectar los mensajes que ya hayan sido recibidos en todos los nodos del clúster; esto se lo realiza debido a que cada nodo debe almacenar los mensajes en caso tenga que retransmitirlo y solo cuando ya exista una confirmación que el mensaje fue visto por todos los nodos, se procede a recolectar y eliminar el mensaje. (Red Hat, Inc., 2013)

pbcast.GMS: Se encarga de unir nuevos nodos al clúster, maneja peticiones de separación del clúster por parte de los nodos y monitorea mensajes para la detección de errores en los protocolos. (Red Hat, Inc., 2013)

UFC: Ajusta la velocidad de envío de mensajes entre los nodos emisor y receptor para que no se presente pérdida de información durante la transmisión de la misma. Esto lo realiza ajustando la velocidad de emisión y recepción a la misma frecuencia. Este protocolo es usado en UNICAST. (Red Hat, Inc., 2013)

MFC: Realiza el mismo trabajo que UFC, pero utiliza MULTICAST. (Red Hat, Inc., 2013)

FRAG2: Fragmenta mensajes muy grandes en mensajes mas pequeños para que puedan ser enviados; el receptor se encarga de volver a unir estos mensajes pequeños y formar el mensaje grande, para así ser entregado a la aplicación. (Red Hat, Inc., 2013)

- Se debe especificar el stack del subsistema de JGroups que va a ser utilizado por defecto. Este stack va a ser utilizado cada vez que se requiera la utilización de un stack pero no se especifica cual. Se utiliza el nombre del stack que fue configurado en el paso anterior.

```
<subsystem xmlns="urn:jboss:domain:jgroups:1.1" default-stack="true-tcp">
```

- Se procede a definir la conexión del JBoss EAP con el servidor que está haciendo el trabajo de balanceador. Se deben añadir dos parámetros en la sección de “mod-cluster-config” de la siguiente forma:

```
<mod-cluster-config advertise-socket="modcluster" proxy-list="IP:puerto"  
balancer="nombre_balanceador" connector="ajp">
```

proxy-list: Lista las direcciones httpd en formato IP: Puerto con la que mod_cluster va a intentar comunicarse

balancer: Es el nombre del balanceador que es definido en el servidor Apache.

- Se debe añadir el parámetro “instance-id” en la declaración del servidor virtual; este nuevo parámetro se lo utiliza para identificar a la instancia de la aplicación (nombre del nodo con el que se va a reconocer). Se utiliza el nombre que se dio en el comienzo del archivo en el server name.

```
<subsystem xmlns="urn:jboss:domain:web:2.1" default-virtual-  
server="default-host" instance-id="{jboss.node.name}" native="false">
```

- Se declara la IP del servidor en donde se va a alojar el JBoss y con la cual se va acceder a la administración web.

```
<interfaces>  
  <interface name="management">  
    <inet-address  
value="{jboss.bind.address.management:192.168.XX.XX}"/>  
  </interface>  
  <interface name="public">  
    <inet-address value="{jboss.bind.address:192.168.XX.XX}"/>  
  </interface>  
  <!-- TODO - only show this if the jacob subystem is added -->  
  <interface name="unsecure">
```

```
<!--  
  ~ Used for IIOp sockets in the standard configuration.  
  ~           To secure JacORB you need to setup SSL  
  -->  
<inet-address  
value="${jboss.bind.address.unsecure:192.168.XX.XX}"/>  
</interface>  
</interfaces>
```

3.10. Beneficios de la implementación de la Guía Metodológica para configurar un Clúster en Alta Disponibilidad para Servidores de Aplicaciones utilizando Open Source

Por la gran competencia que existe actualmente en el mundo informático, es vital que las empresas que ofrecen servicios de soporte puedan ofrecer a sus clientes un funcionamiento de alta calidad en sus aplicaciones. Por esta razón en esta disertación de grado se propone una Guía Metodológica para la Implementación de un Clúster en Alta Disponibilidad para Servidores de Aplicaciones utilizando Open Source para que sea implementada por la empresa Soporte Libre Cía. Ltda. de la ciudad de Quito. Soporte Libre Cía. Ltda. se verá sumamente beneficiada por esta herramienta ya que tendrán una guía clara y concisa a seguir, servirá para poder tomar decisiones adecuadas en el diseño de un clúster con un buen funcionamiento.

Para que esta guía sea implementada con facilidad por parte de la empresa Soporte Libre Cía. Ltda., es necesario tener una definición clara de los conceptos bases que se maneja en esta guía, tales como clúster y sus componentes, servidores de aplicaciones tales como JBoss, y balanceo de carga.

Un clúster es un grupo de servidores independientes (nodos) con características comunes que trabajan en conjunto como un simple recurso integrado. Hay varios tipos de clúster, como Alto Rendimiento, Alta Eficiencia y Alta Disponibilidad, los cuales presentan diferentes características dependiendo la necesidad que tenga la empresa. En el caso de esta disertación de grado, se utilizará un clúster de Alta Disponibilidad ya que este asegura el funcionamiento constante de la aplicación, y prevee que esta aplicación no deje de funcionar. Así mismo los clústeres tienen dos tipos de implementación: Clústeres Activo-Activo y clústeres Activo-Pasivo. En el primer tipo de implementación, los dos

GUÍA METODOLÓGICA PARA LA IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA
DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES, UTILIZANDO OPEN SOURCE.
CASO DE ESTUDIO: EMPRESA SOPORTE LIBRE

Los nodos del clúster funcionan a la par, es decir realizan las mismas tareas simultáneamente. Esto significa que la información siempre se mantiene actualizada en los nodos, sin embargo este tipo de implementación necesita más recursos (hardware, procesamiento, memoria) para que el clúster trabaje a su máxima capacidad. El segundo tipo de implementación es Activo-Pasivo, en el cual un solo nodo trabaja para dar el servicio, mientras los otros nodos se encuentran en reposo, sin ofrecer ningún servicio. A pesar de que se necesita más recursos para la implementación de un clúster Activo-Activo, en esta guía se recomienda que se utilice debido a que brinda mayores beneficios y eficiencia, como el tiempo de respuesta en caso de caída del servicio (failover).

Un servidor de aplicaciones como JBoss, proporciona los recursos necesarios para que la aplicación pueda ser desplegada correctamente en base a su configuración, ya que ofrece Alta Disponibilidad, escalabilidad y fácil mantenimiento. La mayor ventaja de utilizar servidores de aplicaciones es tener una configuración centralizada manteniendo la integridad de datos. Adicionalmente funciona en tres capas en lugar de dos como lo hacen los servidores web. El funcionamiento en tres capas permite estructurar al sistema de manera más simple y eficiente.

Cabe mencionar que los servidores de aplicaciones JBoss utilizan tecnología J2EE, que están basados en Java, lo cual permite tener portabilidad, es decir que se pueda utilizar la aplicación en todas las arquitecturas sin realizar grandes cambios en las configuraciones.

Como se menciono anteriormente, el servidor de aplicaciones que se utiliza en esta guía es JBoss EAP 6.3. Es una plataforma middleware, rápida y segura; provee clustering con alta disponibilidad, mensajería rápida, caché distribuido y otras tecnologías que le ayudan a ser una plataforma estable y escalable.

Como parte de la operatividad del clúster, es importante el balanceador de carga, ya que este permite dividir y canalizar las tareas que son asignadas a un único servidor hacia otros servidores que se encuentran en la misma red. Las características del balanceador evitan la saturación de servicios y gestionan de manera eficiente los recursos, lo que resulta que ningún equipo sea indispensable en el servicio que se ofrece. Un balanceo de carga de software, es adecuado para esta guía metodológica ya que garantiza accesibilidad a cualquier servidor y puede ser configurado para cualquier otra aplicación. El método `mod_cluster` que es basado en `httpd`, utiliza un canal específico de comunicación para reenviar peticiones de acceso a la aplicación. La ventaja de usar este método es que utiliza otro canal para transmitir información del balanceador de carga y realizar un monitoreo constante.

GUÍA METODOLÓGICA PARA LA IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA
DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES, UTILIZANDO OPEN SOURCE.
CASO DE ESTUDIO: EMPRESA SOPORTE LIBRE

La Guía Metodológica para la Implementación de un Clúster en Alta Disponibilidad para Servidores de Aplicaciones utilizando Open Source, se realizó con la visión de que tenga utilidad en una empresa dedicada a la infraestructura informática. Dentro de este marco se eligió a la empresa Soporte Libre Cía. Ltda., ya que brinda ciertas condiciones que permiten que la guía metodológica sea aplicada. Soporte Libre Cía. Ltda. Ofrece una alta calidad en herramientas de Software Libre y Open Source, además de ser Advance Business Partner de Red Hat y JBoss desde el 2007.

Soporte Libre Cía. Ltda. Al utilizar alta disponibilidad en los sistemas que ofrece, tiene una ventaja competitiva en el mercado ya que brinda una rápida accesibilidad a los datos críticos, asegurando así una confianza en los sistemas y en la empresa porque no se compromete la productividad, ingresos y seguridad de los usuarios del sistema.

El propósito de la guía metodológica es proporcionar a Soporte Libre Cía. Ltda., un marco base con una ruta trazada para la implementación de manera estándar un clúster de alta disponibilidad usando Open Source en todos sus sistemas, para que puedan ofrecer a sus clientes el mejor servicio posible.

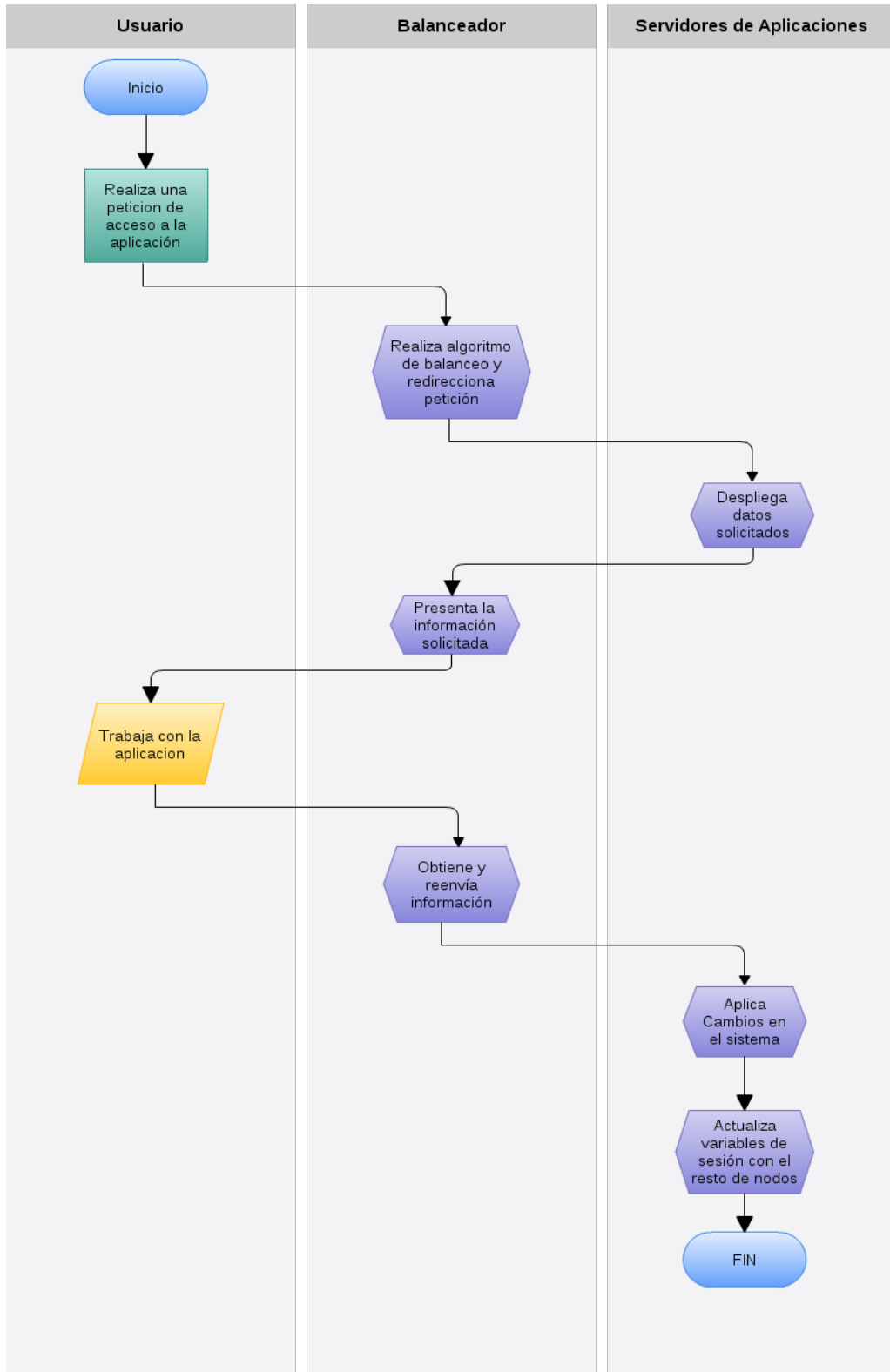
La guía metodológica incluye los siguientes componentes:

- Pasos para la instalación y configuración del Sistema Operativo Red Hat Enterprise Linux 6.6 para el Balanceador y para los Nodos del servidor de aplicaciones JBoss EAP 6.3.
- Configuración del Balanceador: instalación de paquetes requeridos para su funcionamiento; configuraciones necesarias en archivo mod_cluster.conf.
- Configuración del servidor de aplicaciones JBoss EAP 6.3: instalación de paquetes requeridos, configuraciones necesarias para correcto funcionamiento de JBoss en archivo standalone-ha.xml

Para que la guía metodológica tenga una utilidad real se procede realizar su respectiva validación en un ambiente de pruebas, en la cual se aplican todos los conocimientos que se describen en los anteriores capítulos. Se utiliza una aplicación de pruebas.

3.11. Diagrama de Flujo de funcionamiento de clúster en alta disponibilidad JBoss EAP 6.3

Figura 45 Diagrama de Flujo



Fuente: Gabriel Andrade

El proceso inicia cuando el usuario desea visualizar la aplicación. Cuando realiza una petición de acceso a la aplicación, el Balanceador recibe la misma y debe realizar un algoritmo de balanceo, el cual determina que servidor de aplicaciones (nodo) es aquel que esta menos ocupado para utilizarlo, entonces el balanceador reenvía la petición del usuario al servidor de aplicaciones seleccionado. Una vez que el servidor de aplicaciones recibe la petición, procede a desplegar la aplicación al usuario por medio del balanceador como intermediario. De esta forma el usuario percibe que el balanceador es el encargado de desplegar la aplicación.

Con la aplicación ya desplegada, el usuario procede a trabajar en ella y a realizar todas las tareas que necesite. Una vez que termine una actividad en la aplicación, se envía la información para ser actualizada; el balanceador recibe nuevamente y reenvía la petición al servidor de aplicaciones que esta encargado de ese usuario. El servidor de aplicaciones procede a realizar los cambios pertinentes en el sistema y finalmente actualiza las variables de entorno que esta utilizando la aplicación para después compartir con los otros servidores de aplicaciones.

CAPITULO 4

4. VALIDACION DE LA GUIA METODOLOGICA PARA LA IMPLEMENTACION DE UN CLUSTER EN ALTA DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES

Para que la guía metodológica tenga una utilidad real es importante realizar una validación de la misma en una aplicación de prueba, que ya esta configurada para que trabaje con alta disponibilidad. Esto permitirá comprobar la correcta realizacion de la guia metodologica y la adecuada implementacion practica de los conceptos y fundamentos base.

4.1. Implementación de Maquinas Virtuales

De acuerdo a la guia metodológica que se ha realizado las máquinas virtuales tienen las siguientes credenciales de acceso:

Tabla 6 Implementación de máquinas Virtuales para Validación

No	Nombre	Hostname	Dirección IP	Usuario root	Password
1	jboss-nodo1	jbnodo1.gabriel.com	192.168.9.5	root	XXXX
2	Jboss-nodo2	jbnodo2.gabriel.com	192.168.9.6	root	XXXX
3	Apache	apache.gabriel.com	192.168.9.7	root	XXXX

4.2. Instalación de Sistema Operativo Red Hat Enterprise Linux 6.6

Como se va a usar una aplicación sencilla para validar el funcionamiento, se va a utilizar el particionamiento de acuerdo a los requisitos mínimos que se describen en [tabla 3](#) y [tabla 4](#) de esta guía metodológica.

Para los nodos 1 y 2 de JBoss:

Tabla 7 Particionamiento Discos de Nodos 1 y 2 para Validación

Punto de Montaje	Nombre	Tamaño en GB
/boot	boot	500 MB
/	lv_root	5 GB
/swap	swap	2GB
/opt	lv_opt	10 GB
/var	lv_home	1.5 GB
/tmp	lv_tmp	1 GB

Para el Balanceador:

Tabla 8 Particionamiento Discos de Balanceador para Validación

Punto de Montaje	Nombre	Tamaño en GB
/boot	boot	500 MB
/	lv_root	5 GB
/swap	swap	2 GB
/var/log	lv_var_log	1.5 GB
/tmp	lv_tmp	1 GB

4.3. Configuración de nombre del servidor y resolución de nombres

La configuración del nombre del servidor y la resolución de nombres se la realiza de acuerdo a la [tabla 6](#). Se deben realizar los siguientes comandos

```
# vim /etc/sysconfig/network  
  
NETWORK= jbnodo1.gabriel.com  
  
#####  
Para nodo 2 de JBoss  
  
NETWORK= jbnodo2.gabriel.com
```

```
#####
```

```
Para el balanceador
```

```
NETWORK= apache.gabriel.com
```

En los tres servidores que forman el clúster debe ir la misma configuración en la resolución de nombres:

```
# vim /etc/hosts
```

```
192.168.9.5 jbnodo1.gabriel.com jbnodo1
```

```
192.168.9.6 jbnodo2.gabriel.com jbnodo2
```

```
192.168.9.7 apache.gabriel.com apache
```

4.4. Configuración Repositorios

Se realiza la configuración del repositorio por medio del DVD de Red Hat Enterprise Linux 6.6. Referirse al procedimiento indicado en la guía metodológica en el punto 3.2.3.1

```
# mount /dev/cdrom /mnt
```

```
# touch /etc/yum.repos.d/local.repo
```

```
# vim /etc/yum.repos.d/local.repo
```

```
[base]
```

```
name=Local
```

```
baseurl=file:///mnt/Server
```

```
enabled=1
```

```
gpgcheck=1
```

```
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

```
[HighAvailability]
```

```
name=HighAvailability
```

```
baseurl=file:///mnt/HighAvailability
```

```
enabled=1
```

```
gpgcheck=1
```

```
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

```
[LoadBalancer]
name=LoadBalancer
baseurl=file:///mnt/LoadBalancer
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

[ResilientStorage]
name=ResilientStorage
baseurl=file:///mnt/ResilientStorage
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

[ScalableFileSystem]
name=ResilientStorage
baseurl=file:///mnt/ScalableFileSystem
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

Para el balanceador adicionalmente se agregan los siguientes repositorios, tal como se indica en el punto [3.3.1](#)

```
# cd /tmp
# wget http://download.fedoraproject.org/pub/epel/6/x86\_64/epel-release-6-8.noarch.rpm
#yum localinstall epel-release-6-8.noarch.rpm
#yum localinstall --nogpgcheck
http://download1.rpmfusion.org/free/el/updates/6/i386/rpmfusion-free-release-6-1.noarch.rpm
http://download1.rpmfusion.org/nonfree/el/updates/6/i386/rpmfusion-nonfree-release-6-1.noarch.rpm
```

4.5. Configuración Balanceador

Para la configuración del servidor Balanceador se siguen los pasos indicados en el punto [3.3.2.](#)

La configuración queda de la siguiente manera:

```
# vim /etc/httpd/conf.d/mod_cluster.conf

LoadModule slotmem_module modules/mod_slotmem.so
LoadModule manager_module modules/mod_manager.so
LoadModule proxy_cluster_module modules/mod_proxy_cluster.so
LoadModule advertise_module modules/mod_advertise.so

<VirtualHost 192.168.9.7:80>
ManagerBalancerName cluster_tesis

    KeepAliveTimeout 300
    MaxKeepAliveRequests 0
    AdvertiseFrequency 5
    EnableMCPMReceive
    ServerAdvertise Off
    LogLevel debug

    <Location /mod_cluster-manager>
        SetHandler mod_cluster-manager
        Order deny,allow
        Allow from all
    </Location>
</VirtualHost>
```

4.6. Instalación JBoss EAP 6.3

Para la validación de esta guía metodológica se va a descomprimir JBoss EAP 6.3 en el directorio “opt” con el siguiente comando:

```
# unzip jboss-eap-6.3.0.zip
.....
inflating: jboss-eap-6.3/welcome-content/documentation.html
inflating: jboss-eap-6.3/welcome-content/index_noconsole.html
inflating: jboss-eap-6.3/welcome-content/index.html
inflating: jboss-eap-6.3/welcome-content/noconsole.html
inflating: jboss-eap-6.3/LICENSE.txt
inflating: jboss-eap-6.3/JBossEULA.txt
inflating: jboss-eap-6.3/jboss-modules.jar
```

4.7. Instalación Standalone

En el archivo de configuración que se utiliza para el funcionamiento del clúster ,“standalone-ha.xml”, se realizaron los siguientes cambios:

Para el nodo 1 del clúster:

```
<server name="standalone-node1" xmlns="urn:jboss:domain:1.6">
.....
<subsystem xmlns="urn:jboss:domain:jgroups:1.1" default-stack="true-tcp">
<stack name="true-tcp">
    <transport type="TCP" socket-binding="jgroups-tcp"/>
    <protocol type="TCPPING">
        <property name="initial_hosts">
            192.168.9.5[7600],192.168.9.6[7600]
        </property>
    </protocol>
    <protocol type="MERGE2"/>
    <protocol type="FD SOCK" socket-binding="jgroups-tcp-fd"/>
    <protocol type="FD"/>
    <protocol type="VERIFY_SUSPECT"/>
    <protocol type="pbcast.NAKACK"/>
    <protocol type="UNICAST2"/>
    <protocol type="pbcast.STABLE"/>
```

GUÍA METODOLÓGICA PARA LA IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA
DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES, UTILIZANDO OPEN SOURCE.
CASO DE ESTUDIO: EMPRESA SOPORTE LIBRE

```
<protocol type="pbcast.GMS"/>
<protocol type="UFC"/>
<protocol type="MFC"/>
<protocol type="FRAG2"/>
</stack>
.....
<mod-cluster-config advertise-socket="modcluster" proxy-list="192.168.9.7:80"
balancer="cluster_tesis" connector="ajp">
.....
<subsystem xmlns="urn:jboss:domain:web:2.1" default-virtual-server="default-host"
instance-id="{jboss.node.name}" native="false">
<interfaces>
.....
  <interface name="management">
    <inet-address value="{jboss.bind.address.management:192.168.9.5}"/>
  </interface>
  <interface name="public">
    <inet-address value="{jboss.bind.address:192.168.9.5}"/>
  </interface>
  <!-- TODO - only show this if the jacobrb subsystem is added -->
  <interface name="unsecure">
    <!--
    ~ Used for IIOP sockets in the standard configuration.
    ~       To secure JacORB you need to setup SSL
    -->
    <inet-address value="{jboss.bind.address.unsecure:192.168.9.5}"/>
  </interface>
</interfaces>
.....
```

GUÍA METODOLÓGICA PARA LA IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA
DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES, UTILIZANDO OPEN SOURCE.
CASO DE ESTUDIO: EMPRESA SOPORTE LIBRE

Para el nodo 2 del clúster:

```
<server name="standalone-node2" xmlns="urn:jboss:domain:1.6">
.....
<subsystem xmlns="urn:jboss:domain:jgroups:1.1" default-stack="true-tcp">
<stack name="true-tcp">
    <transport type="TCP" socket-binding="jgroups-tcp"/>
    <protocol type="TCPPING">
        <property name="initial_hosts">
            192.168.9.5[7600],192.168.9.6[7600]
        </property>
    </protocol>
    <protocol type="MERGE2"/>
    <protocol type="FD SOCK" socket-binding="jgroups-tcp-fd"/>
    <protocol type="FD"/>
    <protocol type="VERIFY_SUSPECT"/>
    <protocol type="pbcast.NAKACK"/>
    <protocol type="UNICAST2"/>
    <protocol type="pbcast.STABLE"/>
    <protocol type="pbcast.GMS"/>
    <protocol type="UFC"/>
    <protocol type="MFC"/>
    <protocol type="FRAG2"/>
</stack>
.....
<mod-cluster-config advertise-socket="modcluster" proxy-list="192.168.9.7:80"
balancer="cluster_tesis" connector="ajp">
.....
<subsystem xmlns="urn:jboss:domain:web:2.1" default-virtual-server="default-host"
instance-id="{jboss.node.name}" native="false">
<interfaces>
.....
<interfaces>
    <interface name="management">
        <inet-address value="{jboss.bind.address.management:192.168.9.6}"/>
    </interface>
```

```
<interface name="public">
  <inet-address value="{jboss.bind.address:192.168.9.6}"/>
</interface>
<!-- TODO - only show this if the jacobd subsystem is added -->
<interface name="unsecure">
  <!--
    ~ Used for IIOF sockets in the standard configuration.
    ~       To secure JacORB you need to setup SSL
  -->
  <inet-address value="{jboss.bind.address.unsecure:192.168.9.6}"/>
</interface>
</interfaces>
```

4.8. Despliegue de Aplicación

Para la validación de esta guía metodológica se va utilizar una aplicación web de prueba; el trabajo de la aplicación es desplegar las veces que se ha visitado la página en el explorador, y desde que IP se está presentando.

La aplicación está en los dos nodos del clúster en el siguiente directorio:

```
/opt/jboss-eap-6.3/standalone/deployments/
```

A la aplicación se cambió el propietario al usuario “jboss”, ya que este usuario es el encargado del manejo del servidor de aplicaciones JBoss EAP (subir, bajar, reiniciar el servicio)

```
chown jboss:jboss cluster_test.war
```

4.9. Levantamiento de Balanceador

Se levanta el servicio “httpd” en el servidor Balanceador como se lo indica en el punto [3.3.4](#) de la guía metodológica:

```
service httpd start
Iniciando httpd: [ OK ]
# chkconfig httpd on
```

4.10. Levantamiento de Aplicación de prueba

El procedimiento para el levantamiento de la aplicación se sigue el comando que es descrito en el punto [3.7.3.1](#) de la guía metodológica:

Nodo 1

```
# service jboss-eap start
Starting jboss-as: [ OK ]
```

Se verifica en el log si el JBoss se levantó correctamente

```
# tailf /opt/jboss-eap-6.3/standalone/log/server.log

11:47:30,917 INFO [org.infinispan.factories.TransactionManagerFactory]
(ServerService Thread Pool -- 53) ISPN000161: Using a batchMode transaction
manager
11:47:31,238 INFO [org.infinispan.jmx.CacheJmxRegistration] (ServerService
Thread Pool -- 53) ISPN000031: MBeans were successfully registered to the
platform MBean server.
11:47:31,251 INFO [org.infinispan.jmx.CacheJmxRegistration] (ServerService
Thread Pool -- 54) ISPN000031: MBeans were successfully registered to the
platform MBean server.
11:47:31,262 INFO [org.jboss.as.clustering.infinispan] (ServerService Thread Pool -
- 53) JBAS010281: Inició repl caché del contenedor web
11:47:31,266 INFO [org.jboss.as.clustering.infinispan] (ServerService Thread Pool -
- 54) JBAS010281: Inició default-host/cluster_test caché del contenedor web
11:47:31,298 INFO [org.jboss.web] (ServerService Thread Pool -- 53) JBAS018210:
Registrar el contexto web: /cluster_test
11:47:31,632 INFO [org.jboss.as.server] (ServerService Thread Pool -- 28)
JBAS018559: Implementado "cluster_test.war" (runtime-name : "cluster_test.war")
11:47:31,677 INFO [org.jboss.as] (Controller Boot Thread) JBAS015961: Interfaz de
administración http escuchando en http://192.168.9.5:9990/management
11:47:31,678 INFO [org.jboss.as] (Controller Boot Thread) JBAS015951: Consola
de administración escuchando en http://192.168.9.5:9990
11:47:31,678 INFO [org.jboss.as] (Controller Boot Thread) JBAS015874: JBoss
EAP 6.3.0.GA (AS 7.4.0.Final-redhat-19) inició en 20218ms - Inició 225 de 303
servicios (124 servicios son perezosos, pasivos o por demanda)
```

GUÍA METODOLÓGICA PARA LA IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA
DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES, UTILIZANDO OPEN SOURCE.
CASO DE ESTUDIO: EMPRESA SOPORTE LIBRE

Se puede observar el JBoss del Nodo 1 del clúster se levantó correctamente. Se procede a levantar el Nodo 2

Nodo 2

```
# service jboss-eap start
Starting jboss-as: [ OK ]
```

Se verifica en el log si el JBoss se levantó correctamente

```
# tailf /opt/jboss-eap-6.3/standalone/log/server.log

11:53:08,201 INFO [org.jboss.as.clustering] (MSC service thread 1-1) JBAS010238:
Número de miembros del clúster: 2
11:53:08,279 INFO [org.infinispan.factories.TransactionManagerFactory]
(ServerService Thread Pool -- 52) ISPN000161: Using a batchMode transaction
manager
11:53:08,281 INFO [org.infinispan.factories.TransactionManagerFactory]
(ServerService Thread Pool -- 53) ISPN000161: Using a batchMode transaction
manager
11:53:08,570 INFO [org.infinispan.jmx.CacheJmxRegistration] (ServerService
Thread Pool -- 53) ISPN000031: MBeans were successfully registered to the
platform MBean server.
11:53:08,634 INFO [org.infinispan.jmx.CacheJmxRegistration] (ServerService
Thread Pool -- 52) ISPN000031: MBeans were successfully registered to the
platform MBean server.
11:53:08,824 INFO [org.jboss.as.clustering.infinispan] (ServerService Thread Pool -
- 52) JBAS010281: Inició default-host/cluster_test caché del contenedor web
11:53:08,861 INFO [org.jboss.as.clustering.infinispan] (ServerService Thread Pool -
- 53) JBAS010281: Inició repl caché del contenedor web
11:53:08,889 INFO [org.jboss.web] (ServerService Thread Pool -- 53) JBAS018210:
Registrar el contexto web: /cluster_test
11:53:09,369 INFO [org.jboss.as.server] (ServerService Thread Pool -- 28)
JBAS018559: Implementado "cluster_test.war" (runtime-name : "cluster_test.war")
11:53:09,399 INFO [org.jboss.as] (Controller Boot Thread) JBAS015961: Interfaz de
administración http escuchando en http://192.168.9.6:9990/management
11:53:09,400 INFO [org.jboss.as] (Controller Boot Thread) JBAS015951: Consola
de administración escuchando en http://192.168.9.6:9990
11:53:09,400 INFO [org.jboss.as] (Controller Boot Thread) JBAS015874: JBoss
EAP 6.3.0.GA (AS 7.4.0.Final-redhat-19) inició en 13638ms - Inició 225 de 303
servicios (124 servicios son perezosos, pasivos o por demanda)
```

Se puede observar que el JBoss del Nodo 2 del clúster también se levantó correctamente.

Como comprobación final que el JBoss de los nodos 1 y 2 están ya funcionando, se debe revisar nuevamente el log del nodo 1 para verificar que el nodo 2 se unió al clúster.

GUÍA METODOLÓGICA PARA LA IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA
DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES, UTILIZANDO OPEN SOURCE.
CASO DE ESTUDIO: EMPRESA SOPORTE LIBRE

Nodo 1

```
# tailf /opt/jboss-eap-6.3/standalone/log/server.log

11:51:38,058 INFO [org.jboss.as.clustering] (Incoming-1,shared=true-tcp)
JBAS010225: Nueva vista del clúster para la partición web (id: 1, delta: 1, merge:
false) : [standalone-node1/web, jbnodo2/web]
11:51:38,060 INFO [org.infinispan.remoting.transport.jgroups.JGroupsTransport]
(Incoming-1,shared=true-tcp) ISPN000094: Received new cluster view: [standalone-
node1/web|1] [standalone-node1/web, jbnodo2/web]
11:51:53,761 INFO [org.jboss.as.clustering] (Incoming-6,shared=true-tcp)
JBAS010225: Nueva vista del clúster para la partición web (id: 2, delta: -1, merge:
false) : [standalone-node1/web]
11:51:53,762 INFO [org.infinispan.remoting.transport.jgroups.JGroupsTransport]
(Incoming-6,shared=true-tcp) ISPN000094: Received new cluster view: [standalone-
node1/web|2] [standalone-node1/web]
11:53:00,553 INFO [org.jboss.as.clustering] (Incoming-8,shared=true-tcp)
JBAS010225: Nueva vista del clúster para la partición web (id: 3, delta: 1, merge:
false) : [standalone-node1/web, jbnodo2/web]
11:53:00,555 INFO [org.infinispan.remoting.transport.jgroups.JGroupsTransport]
(Incoming-8,shared=true-tcp) ISPN000094: Received new cluster view: [standalone-
node1/web|3] [standalone-node1/web, jbnodo2/web]
```

También se puede verificar por medio del mod_cluster-manager a través de un explorador web se accede al mod_cluster-manager:

```
http://192.168.9.7/mod_cluster-manager
```

Figura 46 Levantamiento de Aplicación de Prueba



Fuente: Resultado de Implementación de Clúster en Alta Disponibilidad utilizando JBoss
EAP 6.3

Resultado:

Los nodos 1 y 2 pertenecen al clúster. Todo esta funcionando correctamente.

4.11. Pruebas de funcionamiento de arquitectura Cluster en Alta Disponibilidad

Se realizan tres pruebas de funcionamiento, en las cuales se verifica que el clúster funciona correctamente. Las pruebas son las siguientes:

4.11.1. Prueba 1

Con los nodos 1 y 2 funcionando se accede a la aplicación que está desplegada en los JBoss.

La direccion es la siguiente:

http://192.168.9.7/cluster_test/

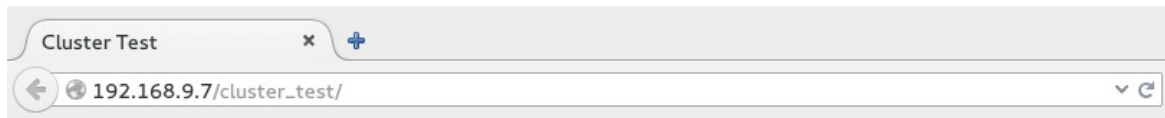
Figura 47 Aplicación desplegada en Prueba Nro. 1



Fuente: Resultado de Ejecución de Clúster en Alta Disponibilidad utilizando JBoss EAP

Como se observa la aplicación esta desplegada; se puede ver el funcionamiento de la aplicación realizando actualizaciones al navegador

Figura 48 Prueba de funcionamiento Aplicación desplegada en Prueba Nro. 1



Cluster Test

You have visited this application 2 times.

This page is being served from 192.168.9.7:80

Fuente: Resultado de Ejecución de Clúster en Alta Disponibilidad utilizando JBoss EAP

6.3

Cabe resaltar que si se utiliza otro navegador el conteo se reiniciará.

4.11.2. Prueba 2

La prueba 2 consiste en el funcionamiento del clúster solo con el nodo 1, se va recrear un ambiente en el que el nodo 2 deje de funcionar. Se procede a bajar el JBoss en el nodo 2.

En el Nodo 2 de JBoss se ejecuta el siguiente comando:

```
# service jboss-eap stop
Parando jboss-as: [ OK ]
```

Se verifica en el log del Nodo 2 que el JBoss esté sin funcionar

```
# tailf /opt/jboss-eap-6.3/standalone/log/server.log

14:41:53,286 INFO [org.jboss.as.connector.subsystems.datasources] (MSC service
thread 1-1) JBAS010409: Fuente de datos desenlazados
[java:jboss/datasources/ExampleDS]
14:41:53,296 INFO [org.apache.coyote.http11.Http11Protocol] (MSC service thread
1-2) JBWEB003075: Coyote HTTP/1.1 pausing on: http-/192.168.9.6:8080
14:41:53,297 INFO [org.apache.coyote.http11.Http11Protocol] (MSC service thread
1-2) JBWEB003077: Coyote HTTP/1.1 stopping on : http-/192.168.9.6:8080
14:41:53,368 INFO [org.jboss.web] (ServerService Thread Pool -- 59) JBAS018224:
Borrar el registro del contexto web: /cluster_test
14:41:53,406 INFO [org.jboss.modcluster] (ServerService Thread Pool -- 56)
```

GUÍA METODOLÓGICA PARA LA IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA
DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES, UTILIZANDO OPEN SOURCE.
CASO DE ESTUDIO: EMPRESA SOPORTE LIBRE

```
MODCLUSTER000021: All pending requests drained from default-host:/cluster_test
in 0,0 seconds
14:41:53,438 INFO [org.jboss.modcluster] (ServerService Thread Pool -- 56)
MODCLUSTER000002: Initiating mod_cluster shutdown
14:41:53,449 INFO [org.infinispan.eviction.PassivationManagerImpl] (ServerService
Thread Pool -- 65) ISPN000029: Passivating all entries to disk
14:41:53,449 INFO [org.infinispan.eviction.PassivationManagerImpl] (ServerService
Thread Pool -- 65) ISPN000030: Passivated 0 entries in 0 milliseconds
14:41:53,453 INFO [org.apache.coyote.ajp] (MSC service thread 1-2)
JBWEB003048: Pausing Coyote AJP/1.3 on ajp-/192.168.9.6:8009
14:41:53,453 INFO [org.apache.coyote.ajp] (MSC service thread 1-2)
JBWEB003051: Stopping Coyote AJP/1.3 on ajp-/192.168.9.6:8009
14:41:53,456 INFO [org.infinispan.eviction.PassivationManagerImpl] (ServerService
Thread Pool -- 56) ISPN000029: Passivating all entries to disk
14:41:53,523 INFO [org.jboss.as.clustering.infinispan] (ServerService Thread Pool -
- 65) JBAS010282: Se detuvo default-host/cluster_test caché del contenedor web
14:41:53,547 INFO [org.jboss.as.server.deployment] (MSC service thread 1-2)
JBAS015877: Se detuvo la implementacióncluster_test.war (runtime-name:
cluster_test.war) en 308ms
14:41:53,556 INFO [org.infinispan.eviction.PassivationManagerImpl] (ServerService
Thread Pool -- 56) ISPN000030: Passivated 1 entries in 99 milliseconds
14:41:53,562 INFO [org.jboss.as.clustering.infinispan] (ServerService Thread Pool -
- 56) JBAS010282: Se detuvo repl caché del contenedor web
14:41:53,566 INFO [org.infinispan.remoting.transport.jgroups.JGroupsTransport]
(MSC service thread 1-1) ISPN000080: Disconnecting and closing JGroups Channel
14:41:53,927 INFO [org.infinispan.remoting.transport.jgroups.JGroupsTransport]
(MSC service thread 1-1) ISPN000082: Stopping the RpcDispatcher
14:41:53,952 INFO [org.jboss.as] (MSC service thread 1-1) JBAS015950: JBoss
EAP 6.3.0.GA (AS 7.4.0.Final-redhat-19) detenido en 666ms
```

En el log del nodo 1 de JBoss se recibe una notificación que es la siguiente:

```
# tailf /opt/jboss-eap-6.3/standalone/log/server.log

14:41:46,418 INFO [org.jboss.as.clustering] (Incoming-15,shared=true-tcp)
JBAS010225: Nueva vista del clúster para la partición web (id: 4, delta: -1, merge:
false) : [standalone-node1/web]
14:41:46,420 INFO [org.infinispan.remoting.transport.jgroups.JGroupsTransport]
(Incoming-15,shared=true-tcp) ISPN000094: Received new cluster view:
[standalone-node1/web|4] [standalone-node1/web]
```

Esta notificación indica que el clúster está funcionando con un solo nodo, que en este caso es el nodo 1. También se puede verificar el funcionamiento del clúster solo con el nodo 1 accediendo al “mod_cluster-manager” a través del explorador

http://192.168.9.7/mod_cluster-manager

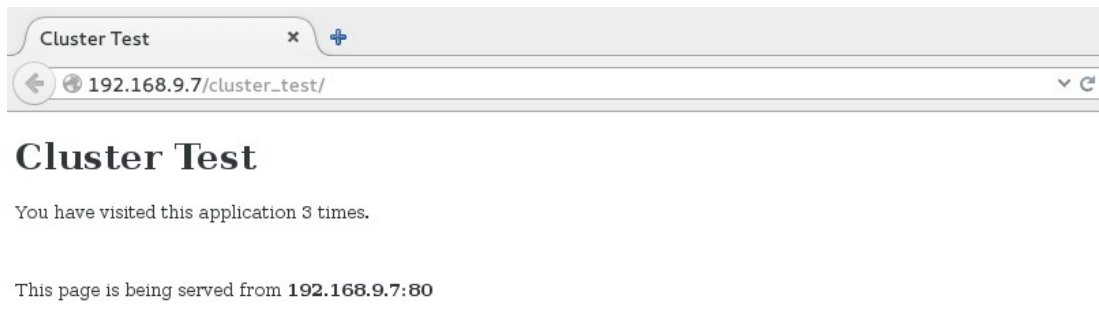
Figura 49 Aplicación desplegada Prueba Nro. 2 – Nodo 1



*Fuente: Resultado de Ejecución de Clúster en Alta Disponibilidad utilizando JBoss EAP
6.3 solo con Nodo 1*

Está verificado que el clúster está solo con el nodo 1, se procede a comprobar que el clúster funciona correctamente. Para esto se va a la aplicación en el explorador y se vuelve a actualizar, el conteo debe seguir aumentando. El conteo anteriormente estaba en **2**.

Figura 50 Prueba Aplicación desplegada Prueba Nro. 2 – Nodo 1



*Fuente: Prueba de Ejecución de Clúster en Alta Disponibilidad utilizando JBoss EAP 6.3
solo con Nodo 1*

Resultado:

El conteo de la aplicación está en **3** lo cual indica que el clúster esta funcionando correctamente solo con el nodo 1.

4.11.3. Prueba 3

La prueba 3 consiste en el funcionamiento del clúster solo con el nodo 2; se va recrear un ambiente en el que el nodo 1 deje de funcionar. Como en la prueba anterior se trabajó solo con el JBoss del nodo 1, el nodo 2 se encuentra abajo, por lo cual se procede a levantar el JBoss en el nodo 2.

Nodo 2

Se ejecuta el siguiente comando:

```
# service jboss-eap start  
Starting jboss-as: [ OK ]
```

Se verifica en el log si el JBoss se levantó correctamente

```
# tailf /opt/jboss-eap-6.3/standalone/log/server.log  
  
16:17:37,854 INFO [org.infinispan.remoting.transport.jgroups.JGroupsTransport]  
(ServerService Thread Pool -- 53) ISPN000094: Received new cluster view:  
[standalone-node1/web]5 [standalone-node1/web, jbnodo2/web]  
16:17:37,857 INFO [org.infinispan.remoting.transport.jgroups.JGroupsTransport]  
(ServerService Thread Pool -- 53) ISPN000079: Cache local address is  
jbnodo2/web, physical addresses are [192.168.9.6:7600]  
16:17:37,865 INFO [org.infinispan.factories.GlobalComponentRegistry]  
(ServerService Thread Pool -- 53) ISPN000128: Infinispan version: Infinispan  
'Delirium' 5.2.10.Final  
16:17:37,906 INFO [org.jboss.as.clustering] (MSC service thread 1-2) JBAS010238:  
Número de miembros del clúster: 2  
16:17:37,954 INFO [org.infinispan.factories.TransactionManagerFactory]  
(ServerService Thread Pool -- 55) ISPN000161: Using a batchMode transaction  
manager  
16:17:37,964 INFO [org.infinispan.factories.TransactionManagerFactory]  
(ServerService Thread Pool -- 53) ISPN000161: Using a batchMode transaction  
manager  
16:17:38,245 INFO [org.infinispan.jmx.CacheJmxRegistration] (ServerService  
Thread Pool -- 55) ISPN000031: MBeans were successfully registered to the  
platform MBean server.  
16:17:38,254 INFO [org.infinispan.jmx.CacheJmxRegistration] (ServerService  
Thread Pool -- 53) ISPN000031: MBeans were successfully registered to the
```

GUÍA METODOLÓGICA PARA LA IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA
DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES, UTILIZANDO OPEN SOURCE.
CASO DE ESTUDIO: EMPRESA SOPORTE LIBRE

platform MBean server.

```
16:17:38,455 INFO [org.jboss.as.clustering.infinispan] (ServerService Thread Pool -  
- 55) JBAS010281: Inició default-host/cluster_test caché del contenedor web  
16:17:38,478 INFO [org.jboss.as.clustering.infinispan] (ServerService Thread Pool -  
- 53) JBAS010281: Inició repl caché del contenedor web  
16:17:38,498 INFO [org.jboss.web] (ServerService Thread Pool -- 53) JBAS018210:  
Registrar el contexto web: /cluster_test  
16:17:39,003 INFO [org.jboss.as.server] (ServerService Thread Pool -- 28)  
JBAS018559: Implementado "cluster_test.war" (runtime-name : "cluster_test.war")  
16:17:39,374 INFO [org.jboss.as] (Controller Boot Thread) JBAS015961: Interfaz de  
administración http escuchando en http://192.168.9.6:9990/management  
16:17:39,375 INFO [org.jboss.as] (Controller Boot Thread) JBAS015951: Consola  
de administración escuchando en http://192.168.9.6:9990  
16:17:39,375 INFO [org.jboss.as] (Controller Boot Thread) JBAS015874: JBoss  
EAP 6.3.0.GA (AS 7.4.0.Final-redhat-19) inició en 14760ms - Inició 225 de 303  
servicios (124 servicios son perezosos, pasivos o por demanda)
```

Se puede verificar que el JBoss se está funcionando correctamente en el nodo 2.

El clúster está funcionando con los nodos 1 y nodo 2, como estaba en el principio. Se debe revisar el log del nodo 1 para verificar que el nodo 2 se unió al clúster.

Nodo 1

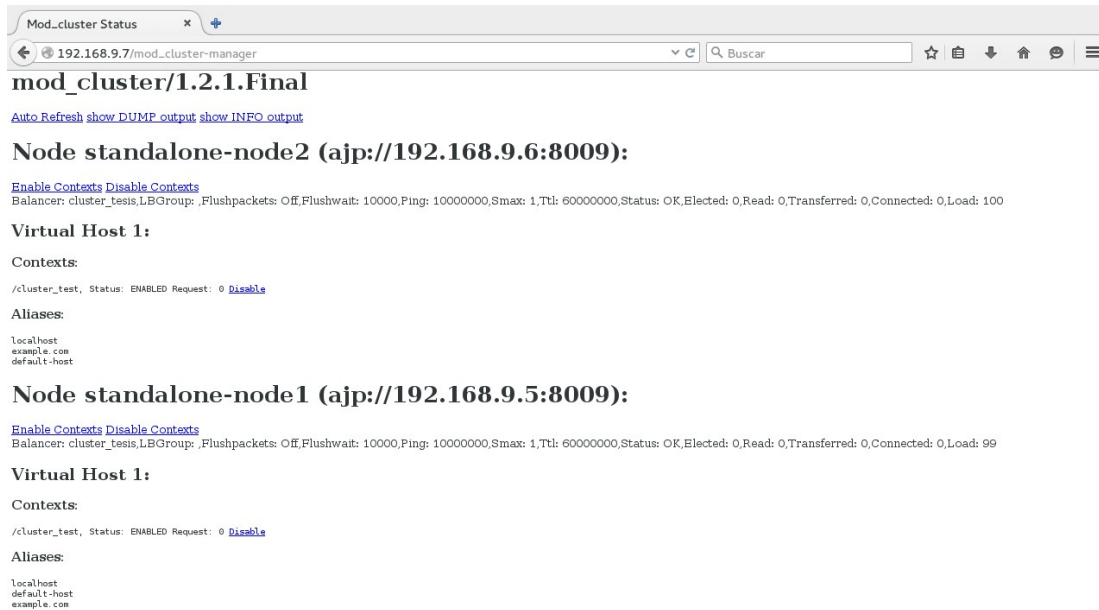
```
# tailf /opt/jboss-eap-6.3/standalone/log/server.log
```

```
16:34:06,873 INFO [org.jboss.as.clustering] (Incoming-1,shared=true-tcp)  
JBAS010225: Nueva vista del clúster para la partición web (id: 1, delta: 1, merge:  
false) : [standalone-node1/web, standalone-node2/web]  
16:34:06,876 INFO [org.infinispan.remoting.transport.jgroups.JGroupsTransport]  
(Incoming-1,shared=true-tcp) ISPN000094: Received new cluster view: [standalone-  
node1/web|1] [standalone-node1/web, standalone-node2/web]
```

Se puede verificar por medio del mod_cluster-manager a través de un explorador web

```
http://192.168.9.7/mod_cluster-manager
```

Figura 51 Nodos 1 y 2 funcionando para Prueba Nro. 3



Fuente: Resultado de Ejecución de Clúster en Alta Disponibilidad utilizando JBoss EAP

6.3

Con los nodos 1 y 2 funcionando se procede a de procede a bajar el JBoss en el nodo 1, para que solo funcione el nodo 2

Nodo 1:

```
# service jboss-eap stop
Parando jboss-as: [ OK ]
```

Se verifica en el log del Nodo 1 que el JBoss esté sin funcionar

```
# tailf log/server.log
17:37:57,823 INFO [org.infinispan.eviction.PassivationManagerImpl] (ServerService Thread Pool -- 65) ISPN000030: Passivated 0 entries in 7 milliseconds
17:37:57,831 INFO [org.jboss.as.clustering.infinispan] (ServerService Thread Pool - - 65) JBAS010282: Se detuvo default-host/cluster_test caché del contenedor web
17:37:57,867 INFO [org.infinispan.eviction.PassivationManagerImpl] (ServerService Thread Pool -- 62) ISPN000030: Passivated 1 entries in 55 milliseconds
17:37:57,873 INFO [org.apache.coyote.ajp] (MSC service thread 1-2) JBWEB003048: Pausing Coyote AJP/1.3 on ajp-/192.168.9.5:8009
17:37:57,874 INFO [org.apache.coyote.ajp] (MSC service thread 1-2) JBWEB003051: Stopping Coyote AJP/1.3 on ajp-/192.168.9.5:8009
17:37:57,890 INFO [org.jboss.as.clustering.infinispan] (ServerService Thread Pool - - 62) JBAS010282: Se detuvo repl caché del contenedor web
17:37:57,893 INFO [org.infinispan.remoting.transport.jgroups.JGroupsTransport] (MSC service thread 1-2) ISPN000080: Disconnecting and closing JGroups Channel
17:37:57,895 INFO [org.jboss.as.server.deployment] (MSC service thread 1-1) JBAS015877: Se detuvo la implementacióncluster_test.war (runtime-name:
```

```
cluster_test.war) en 251ms  
17:37:58,232 INFO [org.infinispan.remoting.transport.jgroups.JGroupsTransport]  
(MSC service thread 1-2) ISPN000082: Stopping the RpcDispatcher  
17:37:58,279 INFO [org.jboss.as] (MSC service thread 1-1) JBAS015950: JBoss  
EAP 6.3.0.GA (AS 7.4.0.Final-redhat-19) detenido en 564ms
```

En el log del nodo 2 de JBoss se recibe una notificación que es el único nodo que está funcionando en el clúster:

```
# tailf log/server.log  
  
17:38:05,640 INFO [org.jboss.as.clustering] (Incoming-7,shared=true-tcp)  
JBAS010225: Nueva vista del clúster para la partición web (id: 2, delta: -1, merge:  
false) : [standalone-node2/web]  
17:38:05,642 INFO [org.infinispan.remoting.transport.jgroups.JGroupsTransport]  
(Incoming-7,shared=true-tcp) ISPN000094: Received new cluster view: [standalone-  
node2/web|2] [standalone-node2/web]
```

Se puede verificar el funcionamiento del clúster solo con el nodo 2 accediendo al “mod_cluster-manager” a través del explorador

http://192.168.9.7/mod_cluster-manager

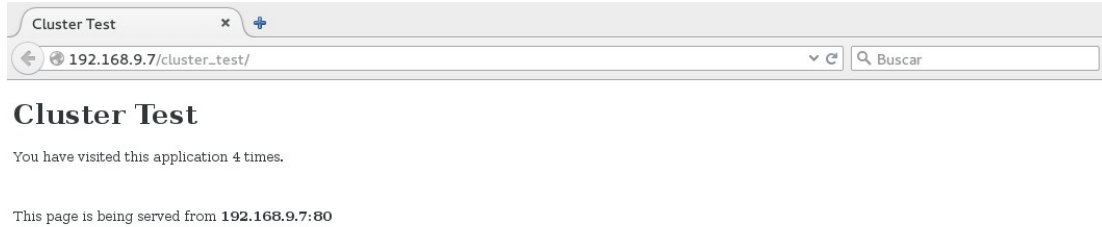
Figura 52 Nodo 2 funcionando para Prueba Nro. 3



*Fuente: Resultado de Ejecución de Clúster en Alta Disponibilidad utilizando JBoss EAP
6.3 solo con Nodo 2*

Se ha verificado que el clúster está solo con el nodo 2, se procede a comprobar que el clúster funciona correctamente. Para esto se va a la aplicación en el explorador y se vuelve a actualizar, el conteo debe seguir aumentando. El conteo anteriormente estaba en 3.

Figura 53 Prueba Aplicación desplegada Prueba Nro. 3 – Nodo 2



Fuente: Prueba de Ejecución de Clúster en Alta Disponibilidad utilizando JBoss EAP 6.3 solo con
Nodo 1

Resultado:

El conteo de la aplicación está en 4 lo cual indica que el clúster esta funcionando correctamente solo con el nodo 2.

CONCLUSIONES

Un servidor de aplicaciones en alta disponibilidad brinda la máxima disponibilidad en todos los servicios, asegurando accesibilidad a los datos a toda hora. Esto es un gran beneficio para los usuarios y también para los desarrolladores porque es un servicio confinable y seguro.

La empresa Soporte Libre Cía. Ltda., ha evidenciado los beneficios al implementar un clúster en alta disponibilidad usando Open Source, ya que de esta manera tendrá la confianza de sus clientes y posteriormente podrá seguir abriendo camino en el mercado Ecuatoriano.

La guía metodológica presentada en esta disertación de grado, se presenta como la herramienta que podrá ser utilizada por Soporte Libre Cía. Ltda. con sus clientes para asegurar una correcta implementación de alta disponibilidad.

1. La herramienta es de fácil implementación porque el servidor de aplicaciones JBoss EAP 6.3 utiliza una configuración centralizada, es decir que todas las configuraciones se realizan en un solo archivo.
2. La guía metodológica utiliza Open Source que es en base a código abierto, teniendo un acceso y control completo a la información, permitiendo hacer todos los cambios según se requiera.
3. Se propone Open Source ya que su característica principal es ser código abierto, lo cual permite tener acceso al código fuente, y facilita la obtención, acceso e intercambio de información y documentación.²
4. La guía utiliza JBoss que tiene servicio Activo-Activo, lo cual garantiza que la información se mantiene actualizada compartiendo las variables de sesión, es decir que si llega a existir un fallo no se pierde el estado en el que se encuentra la información.
5. Sólo se requieren cambios mínimos en las configuraciones, ya que el archivo utilizado para alta disponibilidad ya viene con los módulos de alta disponibilidad cargados por defecto, sólo se necesita realizar leves cambios para adecuarse al objetivo de uso del servidor de aplicaciones.
6. Al utilizar JBoss como servidor de aplicaciones, que está basado en tecnología JAVA, permite tener portabilidad, permitiendo utilizar el servidor de aplicaciones

² Esta afirmación está basada en la experiencia profesional del autor de esta disertación de grado, lo cual ha permitido corroborar los beneficios de utilizar Open Source para servidores de aplicaciones

GUÍA METODOLÓGICA PARA LA IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA
DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES, UTILIZANDO OPEN SOURCE.
CASO DE ESTUDIO: EMPRESA SOPORTE LIBRE

en todas las arquitecturas ya sean open Source o privadas como Windows. En base a esto cabe mencionar que esta guía metodológica no solo sirve para la tecnología propuesta pero para una variedad de herramientas lo cual hace que esta guía metodológica sea versátil y con un gran alcance de aplicación.

7. También la guía metodológica tiene una gran utilidad porque se la puede utilizar para implementar clústeres en alta disponibilidad en varios tipos de aplicaciones web. Esto permite que la guía sea usada en varios clientes siempre y cuando cuenten con todos los requisitos necesarios.
8. La guía metodológica permite una rápida implementación en los clientes, facilitando el trabajo de los técnicos en Soporte Libre Cia. Ltda.

RECOMENDACIONES

1. Para un correcto funcionamiento de la aplicación, tanto el servidor de aplicaciones como la aplicación como tal deben estar configuradas para alta disponibilidad.
2. La velocidad depende del servidor el cual aloja al servidor de aplicaciones por lo que se recomienda un servidor adecuado de acuerdo a las actividades de la aplicación.
3. El administrador del servidor de aplicaciones debe tener conocimientos básicos de GNU LINUX para que pueda manejar todos los componentes y se le facilite la implementación de la presente guía y el mantenimiento.
4. Se recomienda la utilización de una partición independiente al Sistema Operativo que aloje la instalación del Servidor de Aplicaciones JBoss EAP 6.3, la partición que se recomienda utilizar es “/opt”
5. Se recomienda por seguridad que siempre el firewall tanto en los servidores de aplicaciones JBoss EAP 6.3 como en el Balanceador estén activados y sólo con los puertos que se necesiten ser usados.
6. Para que no exista problemas de conectividad entre el Balanceador y los servidores de aplicaciones JBoss EAP 6.3, se recomienda que todos los servidores se encuentren en el mismo segmento de red.
7. Por razones de seguridad es muy recomendable mantener a los servidores Red Hat Enterprise Linux actualizados.
8. Se recomienda que los servidores de producción que alojan a los servidores de Aplicaciones JBoss EAP 6.3 dispongan de un mínimo de 100GB de disco local y de 100GB para la partición que se utilizará para JBoss EAP 6.3, también para un mejor funcionamiento al menos 16GB de RAM.
Para el Balanceador se recomienda que se disponga de un mínimo de 100GB de disco local y de 16 GB de RAM.
9. El siguiente paso para mejorar la seguridad y para aumentar la confiabilidad es la implementación de certificados SSL, por ello se recomienda la utilización de una entidad certificadora para la generación de estos certificados. Con la utilización de certificados SSL se podrá mejorar el servicio al cliente ofreciendo un producto que cumpla con los estándares de seguridad deseados.
10. Actualmente es sumamente importante al momento de diseñar la infraestructura contar con un sistema que ofrezca al usuario confianza y estabilidad. La Alta

Disponibilidad cumple con estas características por lo que recomiendo que este conocimiento sea incluido dentro del pensum académico de la Facultad de Ingeniería en Sistemas de la Universidad Católica del Ecuador. Con este conocimiento los estudiantes tendrían la capacidad involucrarse de mejor manera al mercado laboral.

11. Mi recomendación para la facultad de Ingeniería en Sistemas en la Universidad Católica del Ecuador, es la implementación de materias orientadas a Open Source, empezando desde la línea base que es el Sistema Operativo. Considero que es muy importante el conocimiento de todas las herramientas existentes en el mercado ya que eso daría a los estudiantes una ventaja al momento de involucrarse en el mundo laboral.
12. También yo considero necesario que se enseñen nuevas tecnologías como por ejemplo cloud, virtualización y herramientas de monitoreo de aplicaciones. Es ideal que existan laboratorios físicos en la facultad con todo el equipamiento necesario, sin embargo en caso de que no se pueda tener laboratorios físicos en la facultad, se incentive la investigación a los estudiantes en el uso de nuevas herramientas y nuevas tecnologías.

ÍNDICE DE IMÁGENES.

Figura 1 Clúster Activo-Activo.....	7
Figura 2 Clúster Activo-Pasivo	8
Figura 3 Componentes de un clúster.....	9
Figura 4 Organigrama Institucional Soporte Libre Cía. Ltda	21
Figura 5 Planificación Estratégica Soporte Libre Cía. Ltda.....	22
Figura 6 Planificación Estratégica Soporte Libre Cía. Ltda.....	23
Figura 7 Diagrama de Infraestructura	25
Figura 8 Configuración Plataforma virtualización KVM Nro.1.....	29
Figura 9 Configuración Plataforma virtualización KVM Nro. 2.....	29
Figura 10 Configuración Plataforma virtualización KVM Nro. 3.....	30
Figura 11 Selección de Medio de Instalación.....	30
Figura 12 Asignación de Memoria.....	31
Figura 13 Asignación de Espacio de Almacenamiento.....	32
Figura 14 Configuración interfaces de red y nombre	33
Figura 15 Menú de Arranque	34
Figura 16 Comprobación Medio de Instalación	35
Figura 17 Inicio de Instalación	35
Figura 18 Inicio de Instalación	36
Figura 19 Selección de teclado	37
Figura 20 Dispositivos de Almacenamiento Parte 1	38
Figura 21 Dispositivos de Almacenamiento Parte 2	38
Figura 22 Configuración de nombre de host.....	39
Figura 23 Configuración de huso horario.....	39
Figura 24 Establecimiento de contraseña de usuario root.....	40
Figura 25 Configuración Particionamiento del Disco.....	42
Figura 26 Configuración Particionamiento del Disco Nodo 1 y 2	42
Figura 27 Configuración Particionamiento del Disco Balanceador	43
Figura 28 Confirmación de Configuración Particionamiento del Disco	43
Figura 29 Configuración del Gestor de Arranque	44
Figura 30 Opciones de Tipo de Servidor.....	46
Figura 31 Opciones de Tipo de Servidor.....	47
Figura 32 Opción de Paquetes Opcionales.....	48
Figura 33 Opción de Paquetes Base.....	48
Figura 34 Selección de Tipo de Servidor	49
Figura 35 Finalización de Instalación Red Hat Enterprise Linux 6.6	50
Figura 36 Inicio de Registro a Red Hat Network.....	54
Figura 37 Ingreso de usuario y contraseña	55
Figura 38 Selección de Versión de Sistema Operativo	55
Figura 39 Confirmación de actualizaciones.....	56
Figura 40 Nombre del perfil de Servidor.....	56
Figura 41 Confirmación de registro de paquetes instalados en servidor.....	57
Figura 42 Finalización de registro a Red Hat Network.....	57
Figura 43 Envío de información a Red Hat Network	58
Figura 44 Confirmación de suscripción a Red Hat Network.....	58
Figura 45 Diagrama de Flujo	80
Figura 46 Levantamiento de Aplicación de Prueba.....	93

GUÍA METODOLÓGICA PARA LA IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA
DISPONIBILIDAD PARA SERVIDORES DE APLICACIONES, UTILIZANDO OPEN SOURCE.
CASO DE ESTUDIO: EMPRESA SOPORTE LIBRE

Figura 47 Aplicación desplegada en Prueba Nro. 1.....	94
Figura 48 Prueba de funcionamiento Aplicación desplegada en Prueba Nro. 1	95
Figura 49 Aplicación desplegada Prueba Nro. 2 – Nodo 1.....	97
Figura 50 Prueba Aplicación desplegada Prueba Nro. 2 – Nodo 1	97
Figura 51 Nodos 1 y 2 funcionando para Prueba Nro. 3.....	100
Figura 52 Nodo 2 funcionando para Prueba Nro. 3.....	101
Figura 53 Prueba Aplicación desplegada Prueba Nro. 3 – Nodo 2	102

ÍNDICE DE TABLAS.

Tabla 1 Equipos Virtuales para el entorno de un clúster en Alta Disponibilidad.....	26
Tabla 2 Configuración de red para las máquinas en el entorno de un clúster de Alta Disponibilidad.....	26
Tabla 3 Particionamiento del Sistema Operativo para Nodo de JBoss	27
Tabla 4 Particionamiento del Sistema Operativo para Apache	28
Tabla 5 Niveles de notificación mod_cluster	61
Tabla 6 Implementación de máquinas Virtuales para Validación	82
Tabla 7 Particionamiento Discos de Nodos 1 y 2 para Validación.....	83
Tabla 8 Particionamiento Discos de Balanceador para Validación.....	83