Red Hat Enterprise Linux 6

Visión general de adición de alta disponibilidad

Sinopsis de la adición de alta disponibilidad para Red Hat Enterprise Linux

Edición 6

Last Updated: 2017-10-29

Red Hat Enterprise Linux 6 Visión general de adición de alta disponibilidad

Sinopsis de la adición de alta disponibilidad para Red Hat Enterprise Linux
Edición 6

Legal Notice

Copyright © 2014 Red Hat, Inc. and others.

This document is licensed by Red Hat under the Creative Commons Attribution-ShareAlike 3.0 Unported License. If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Resumen

Visión general de la adición de alta disponibilidad brinda un sinopsis de la adición de alta disponibilidad para Red Hat Enterprise Linux 6.

Table of Contents

INTRODUCCIÓN

Este documento proporciona una visión general de la adición de alta disponibilidad para Red Hat Enterprise Linux 6.

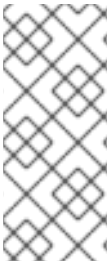
Para el buen entendimiento de este documento, se deberá tener una amplia experiencia en Red Hat Enterprise Linux y entender los conceptos de computación de servidor.

Para obtener más información sobre el uso de Red Hat Enterprise Linux, consulte los siguientes recursos:

- *Guía de instalación de Red Hat Enterprise Linux*— Proporciona información sobre la instalación de Red Hat Enterprise Linux 6.
- *Guía de implementación de Red Hat Enterprise Linux*— Proporciona información sobre la implementación, configuración y administración de Red Hat Enterprise Linux 6.

Para obtener más información sobre este producto y los productos relacionados a Red Hat Enterprise Linux 6, consulte los siguientes recursos:

- *Cómo configurar y administrar la adición de alta disponibilidad*Proporciona información sobre la configuración y manejo de la adición de alta disponibilidad (conocida también como Red Hat Cluster) para Red Hat Enterprise Linux 6.
- *Administración del Gestor de volúmenes lógicos*— Proporciona una descripción del Gestor de Volúmenes Lógicos (LVM) e incluye información sobre la ejecución de LVM en un entorno de clúster.
- *Sistema de archivos global 2: Configuración y administración*— Proporciona información sobre la instalación, configuración y mantenimiento de Red Hat GFS2 (Red Hat Global File System 2), el cual se incluye en la adición de almacenamiento resistente.
- *DM Multipath* — Proporciona información sobre el uso de la función Device-Mapper Multipath de Red Hat Enterprise Linux 6.
- *Equilibrador de cargas de Red Hat*— Proporciona información sobre cómo configurar sistemas y servicios de alto rendimiento con la adición del Equilibrador de cargas de Red Hat (Conocido anteriormente como Servidor Virtual de Linux [LVS]).
- *Notas de lanzamiento*— Proporciona información sobre el lanzamiento actual de productos Red Hat.



NOTA

Para obtener información sobre las mejores prácticas para implementar y actualizar los clústeres de Red Hat Enterprise Linux mediante la adición de alta disponibilidad y el Sistema de archivos globales 2 de Red Hat (GFS2), consulte el artículo "Red Hat Enterprise Linux Cluster, High Availability, y GFS Deployment Best Practices" en Red Hat Customer Portal, . https://access.redhat.com/kb/docs/DOC-40821.

Este y otros documentos de Red Hat están disponibles en versiones HTML, PDF, y RPM en el CD de documentación de Red Hat Enterprise Linux y en línea en http://access.redhat.com/documentation/docs.

1. ¡NECESITAMOS SUS COMENTARIOS!

Si encuentra algún error tipográfico o si tiene alguna sugerencia para mejorar este documento, nos gustaría saberlo. Por favor complete un reporte en Bugzilla: http://bugzilla.redhat.com/ con el producto **Red Hat Enterprise Linux 6** , el componente *doc-High_Availability_Add-On_Overview* y el número de versión: **6 . 6**.

Si tiene alguna sugerencia para mejorar la documentación, trate de ser lo más específico posible al describirla. Si ha encontrado algún error, por favor incluya el número de la sección y parte del contexto para que sea más fácil encontrarlo.

CAPÍTULO 1. VISIÓN GENERAL DE ADICIÓN DE ALTA DISPONIBILIDAD

La adición de alta disponibilidad es un sistema en clúster que ofrece fiabilidad, escalabilidad y disponibilidad a servicios de producción crítica Las secciones a continuación proporcionan una descripción de los componentes y funciones de alta disponibilidad de alto nivel:

- Sección 1.1, “Fundamentos de clúster”
- Sección 1.2, “Introducción a la Adición de alta disponibilidad ”
- Sección 1.3, “Infraestructura de clúster”

1.1. FUNDAMENTOS DE CLÚSTER

Un clúster está compuesto por dos o más computadores (llamados *nodos* o *miembros*) que trabajan juntos para ejecutar una tarea. Hay cuatro clases de clúster:

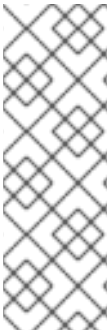
- Almacenamiento
- Alta disponibilidad
- Balance de carga
- Alto rendimiento

Los clústeres de almacenamiento proporcionan una imagen de sistema de archivos consistente, a través de los servidores en el clúster, lo cual permite que los servidores lean y escriban simultáneamente un sistema de archivos compartido. Un clúster de almacenamiento simplifica la administración de almacenamiento al limitar la instalación de aplicaciones a un sistema de archivos. Asimismo, con un sistema de archivos a lo largo del clúster, un clúster de almacenamiento elimina la necesidad de copias de más de los datos de la aplicación y simplifica la creación de copias de seguridad y recuperación contra desastres. La adición de alta disponibilidad proporciona agrupamiento de almacenamiento junto con Red Hat GFS2 (parte de la adición de almacenamiento resistente).

Los clústeres de alta disponibilidad proporcionan alta disponibilidad de servicios mediante la eliminación de los puntos individuales de falla y la conmutación de servicios de un nodo de clúster a otro en caso de que un nodo sea inoperante. Generalmente, los servicios en un clúster de alta disponibilidad leen y escriben datos (a través de los sistemas de archivos de lectura y escritura montados). Así, un clúster de alta disponibilidad debe mantener la integridad de los datos cuando un nodo reciba el control del servicio desde otro nodo de clúster. Las fallas de nodos en un clúster de alta disponibilidad no están visibles para los clientes externos al clúster. (Los clústeres de alta disponibilidad también se conocen como clústeres de conmutación.) La adición de alta disponibilidad proporciona agrupamiento a través de su componente de administración de servicios de alta disponibilidad, `rgmanager`..

Los clústeres de balance de carga responden a peticiones de servicios de red desde diferentes nodos para balancear las peticiones a los largo de los nodos del clúster. El balance de carga proporciona escalabilidad económica porque se puede configurar el número de nodos de acuerdo con los requerimientos de balance de carga. Si un nodo en un clúster de balance de carga falla, el software de balance de carga detecta la falla y asigna las peticiones a otros nodos en el clúster. Los nodos erróneos en un clúster de balance de carga no son visibles desde los clientes fuera del clúster.

Los clústeres de alto rendimiento utilizan los nodos para ejecutar cálculos simultáneos. Un clúster de alto rendimiento permite que las aplicaciones trabajen de forma paralela, mejorando así el rendimiento de éstas. El clúster de alto rendimiento se conoce como clúster computacional o computación en red.



NOTA

Los tipos de cluster resumidos anteriormente reflejan las configuraciones básicas. Según las necesidades del usuario, se podría requerir de una combinación de los clústeres descritos.

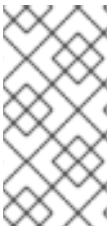
Además, Red Hat Enterprise Linux High Availability Add-On contiene soporte para configurar y administrar servidores de alta disponibilidad *únicamente. No ofrece soporte* para clústeres de alto rendimiento.

1.2. INTRODUCCIÓN A LA ADICIÓN DE ALTA DISPONIBILIDAD

La adición de alta disponibilidad es un conjunto integrado de componentes de software que puede ser implementado en una amplia variedad de configuraciones para cubrir la necesidad de rendimiento, alta disponibilidad, balance de carga, escalabilidad, compartición de archivos y economía de recursos.

La adición de alta disponibilidad consta de los siguientes componentes principales:

- Infraestructura de clúster — proporciona funciones fundamentales para que los nodos trabajen juntos como un clúster: administración del archivo de configuración, administración de membresías, administración de cierres de exclusión y cercado.
- La administración de servicios de alta disponibilidad — Proporciona la transferencia de servicios de un clúster a otro en caso de que un nodo falle.
- La herramientas de administración de clúster — Herramientas de configuración y administración para configurar y administrar una adición de alta disponibilidad. Las herramientas se utilizan con los componentes de infraestructura de clúster, los componentes de alta disponibilidad, de administración de servicios y de almacenamiento.



NOTA

Únicamente los clústeres de un solo sitio son compatibles en este momento. Los clústeres esparcidos a través de varios lugares físicos no tienen soporte formal. Si desea obtener más información sobre clústeres multisitios, por favor contacte a su representante de soporte técnico de Red Hat.

Puede complementar la adición de alta disponibilidad con los siguientes componentes:

- Red Hat GFS2 (el sistema de archivos globales 2); Parte de la adición de almacenamiento resistentes, que provee un sistema de archivos de clúster para usar con la adición de alta disponibilidad. GFS2 acepta múltiples nodos para compartir almacenamiento en un nivel de bloque como si el almacenamiento estuviera conectado de forma local a un nodo de clúster . Un sistema de archivos de clúster GFS2 requiere una infraestructura de clúster.
- El Gestor de volúmenes lógicos de clúster (CLVM) — Parte de la adición de almacenamiento resistente, la cual proporciona administración de volúmenes de almacenamiento de clústeres. CLVM también soporta infraestructura de clúster.
- La adición del balanceador de carga — software de enrutamiento que proporciona balance de carga IP. La adición de balance de carga se ejecuta en un par de servidores virtuales

redundantes que distribuye uniformemente las solicitudes de clientes a los servidores reales que están detrás de los servidores virtuales.

1.3. INFRAESTRUCTURA DE CLÚSTER

La infraestructura de clúster de alta disponibilidad proporciona las funciones básicas para que un grupo de computadores (llamados *nodos* o *miembros*) trabajen juntos en un clúster. Una vez el clúster ha sido formado con la infraestructura de clúster, puede utilizar otros componentes para cubrir las necesidades de agrupamiento (por ejemplo, puede establecer un clúster para compartir archivos en un sistema de archivo GFS2 o establecer un servicio de conmutación). La infraestructura de clúster lleva a cabo las siguientes funciones:

- Administración de cluster
- Administración de los cierres de exclusión
- Cercado
- Administración de la configuración de cluster

CAPÍTULO 2. ADMINISTRACIÓN DE CLÚSTER CON CMAN

La administración de clúster maneja el cuórum y la membresía de clúster. CMAN (abreviación en inglés del Gestor de clústeres) administra los clústeres en la adición de alta disponibilidad para Red Hat Enterprise Linux. CMAN es un gestor de clústeres distribuido que se ejecuta en cada nodo de clúster; la administración de clústeres se distribuye a través de todos los nodos en el clúster.

CMAN guarda el rastro de las membresías al monitorizar los mensajes desde otros nodos de clúster. Cuando las membresías de clúster cambian, el gestor de clúster notifica a los otros componentes de la infraestructura para que lleven a cabo las acciones apropiadas. Si un nodo de clúster no transmite un mensaje durante un tiempo determinado, el gestor de clúster retira el nodo del clúster y comunica a los otros componentes de la infraestructura de clúster que el nodo ya no es miembro del clúster.

CMAN realiza un seguimiento del cuórum del clúster sondeando la cuenta de nodos de clúster. Si más de la mitad de los nodos están activos, el clúster tiene cuórum. Si la mitad o menos de la mitad de los nodos están activos, no hay cuórum en el clúster y toda la actividad de clúster se detiene. El cuórum del clúster evita que se presente la condición de cerebro dividido o "split-brain" — una condición en la cual dos instancias del mismo clúster se ejecutan al mismo tiempo. Esta condición permitiría que cada instancia de clúster accediera a los recursos sin conocimiento de la otra instancia de clúster, lo cual produciría la corrupción de la integridad del clúster.

2.1. CUÓRUM DE CLÚSTER

Cuórum es un algoritmo de votación utilizado por CMAN.

Un clúster solo puede funcionar correctamente si hay un acuerdo general entre los miembros en relación con sus estatus. Decimos que un clúster tiene cuórum si la mayoría de nodos está viva, comunicándose y en acuerdo con los miembros de clúster activos. Por ejemplo, en un clúster de trece nodos, el cuórum solo se alcanza si siete o más nodos se están comunicando. Si el séptimo nodo muere, el clúster pierde cuórum y no puede funcionar más.

Un clúster debe mantener el cuórum para evitar problemas *split-brain* . Si el cuórum no era obligatorio, el cuórum, un error de comunicación en ese mismo clúster de trece nodos puede causar una situación en la que seis nodos operen en el almacenamiento compartido , mientras que otros seis nodos también operen en él independientemente. Debido al error de comunicación, los dos clústeres parciales sobrescribirían las áreas del disco y dañarían el sistema de archivos. Con las reglas de cuórum obligatorias, solamente uno de los clústeres parciales puede usar el almacenamiento compartido, lo cual protege la integridad de los datos.

El cuórum no evita situaciones de cerebro dividido, porque no decide quién el dominante y puede funcionar en el clúster. En caso de presentarse la situación de cerebro dividido, el cuórum impide al grupo de clústeres hacer otra cosa.

El Cuórum es determinado por la comunicación de mensajes entre los nodos de clúster vía Ethernet. También, se puede determinar por una combinación de mensajes de comunicación a través de Ethernet *y* de un disco de cuórum. Para el cuórum vía Ethernet, el cuórum consta de una mayoría simple (50 % de los nodos + 1 adicional). Cuando se configura un disco de cuórum, el cuórum consta de condiciones de usuario especificadas.

NOTA

Por defecto, cada nodo tiene un voto. Sin embargo, se puede modificar la configuración para que cada nodo tenga más de un voto.

2.1.1. Discos de cuórum

Un disco de cuórum o partición es una sección de un disco que está configurado para se usado con los componentes del proyecto de clúster. Tiene varios propósitos. Por ejemplo:

Suponga que tiene los nodos A y B, y el nodo A no puede obtener varios de los paquetes "heartbeat" del gestor de clústeres. El nodo A no sabe por qué no ha recibido los paquetes, pero hay varias posibilidades: el nodo B ha fallado, el interruptor de red o concentrador ha fallado, el adaptador de red de nodo A ha fallado o quizás se debió a que el nodo B estaba demasiado ocupado para enviar el paquete. Esto puede suceder si el clúster es demasiado grande, sus sistemas están muy ocupados o si su red es poco fiable.

El nodo A no sabe cuál es el caso y desconoce si el problema reside en él o en el nodo B. Esto es problemático especialmente en un clúster de dos nodos, porque estos dos nodos no se tocan entre sí y cada uno puede tratar de cercar al otro.

Por lo tanto, antes de cercar un nodo, sería conveniente tener otra forma de verificar si el otro nodo está realmente vivo, a pesar de que no parezcan estar en contacto. Un disco de cuórum le da la destreza para hacer justo esto. Antes de cercar un nodo que esté fuera de contacto, el software de clúster puede verificar si el nodo está aún vivo basándose en si ha escrito datos en la partición de cuórum o no.

En el caso de sistemas de dos nodos, el disco de cuórum también actúa como un disyuntor. Si un nodo ha ingresado al disco de cuórum y la red, eso cuenta como dos votos.

Un nodo que haya perdido contacto con la red o disco de cuórum pierde el voto, y por lo tanto, es conveniente cercarlo por seguridad.

Para obtener más información sobre configuración de los parámetros de disco de cuórum, consulte los capítulos sobre administración de Conga y **ccs** en la guía *Administración de clústeres*.

2.1.2. Tie-breakers (Disyuntores)

Los disyuntores son heurística adicional que permiten a una partición de clúster decidir si hay cuórum (quorate) o no en el evento de una división equitativa anterior al cercado. Un disyuntor típico es un disyuntor IP, conocido también como nodo 'ping'.

Con estos disyuntores, los nodos no solo se controlan entre sí, sino también controlan al enrutador principal que está en la misma ruta de comunicaciones de clúster. Si los dos nodos pierden contacto entre sí, el ganador es el que aún pueda contactar al enrutador principal. Claro está, que en casos tales como en un interruptor de bucle, donde los dos nodos pueden ver al enrutador, pero no verse entre sí, se produce un cerebro dividido, incluso cuando se utilizan disyuntores. Por esta razón, es importante asegurarse de que el cercado esté configurado correctamente.

Otros tipos de disyuntores incluyen dónde la partición compartida, la cual suele llamarse disco de cuórum, proporciona información adicional. clumanager 1.2.x (Red Hat Cluster Suite 3) tenía un disyuntor de discos que permitía operar si la red estaba caída, siempre y cuando ambos nodos aún estuvieran comunicándose en la partición compartida.

Existen esquemas de disyuntores más complejos, tales como QDisk (parte de linux-cluster). QDisk permite especificar la heurística arbitraria. Esta permite que cada nodo determine su propia estabilidad de participación en el clúster. No obstante, suele usarse como un simple disyuntor IP. Para obtener más información, consulte la página de manual qdisk(5).

Cman no presenta disyuntos internos por varias razones. No obstante, los disyuntores pueden implementarse mediante la API. Esta API permite registro y actualización de dispositivos de cuórum. Por ejemplo, observe el código fuente de QDisk.

Será necesario usar un disyuntor si:

- Tiene una configuración de dos nodos con dispositivos de cercado en una ruta de red diferente a la ruta utilizada para comunicación de clúster.
- Tiene una configuración de dos nodos en donde el cercado está en el nivel de fábrica - especialmente para reservaciones SCSI.

No obstante, si la red y la configuración de cercado en su clúster están correctas, un disyuntor solo añade complejidad, salvo en casos excepcionales.

CAPÍTULO 3. RGMANAGER

RGManager administra y proporciona las capacidades de conmutación para conjuntos de recursos de clúster llamados servicios, grupos de recursos o árboles de recursos. Estos grupos de recursos se estructuran en forma de árbol y tienen dependencia de padre-hijo y relaciones de herencia dentro de cada subárbol.

RGManager permite a los administradores definir, configurar y monitorizar los servicios de clúster. En el evento de que un nodo falle, RGManager reasignará el servicio en clústeres a otro nodo con un mínima interrupción del servicio. Usted puede restringir servicios a ciertos nodos, tales como **`httpd`** a un grupo de nodos, mientras que **`mysql`** puede restringirse a un conjunto independiente de nodos.

Hay varios procesos y agentes que se combinan para que RGManager funcione. La siguiente lista resume esas áreas:

- Dominios de conmutación - Cómo funciona el sistema de dominio de conmutación de RGManager
- Políticas de servicios - Inicio de servicio de Rgmanager y políticas de recuperación
- Árboles de recursos - Cómo funcionan los árboles de recursos de rgmanager, incluidas órdenes de inicio/detención y herencia
- Conductas operativas de servicios - Cómo funcionan las operaciones de rgmanager y lo que significan los estados
- Conductas de máquinas virtuales - Cosas especiales para recordar al ejecutar máquinas virtuales en un clúster rgmanager
- Acciones de recursos - Las acciones que el agente RGManager utiliza y la forma de personalizar su conducta desde el archivo **`cluster.conf`** .
- Scripting de eventos - Si las políticas de conmutación y recuperación no se ajustan a su entorno, usted puede personalizar su propio entorno mediante el subsistema de scripting.

3.1. DOMINIOS DE CONMUTACIÓN

Un dominio de conmutación es un subconjunto de miembros ordenados a los cuales se puede vincular un servicio. Aunque . Los dominios de conmutación son útiles para personalizar el clúster, no se requieren para funcionar.

La lista a continuación presenta la semántica que gobierna las opciones de la forma como las diferentes opciones de configuración afectan la conducta de un dominio de conmutación.

- El nodo o el miembro preferido: El nodo preferido era el miembro designado para ejecutar el servicio si el miembro estaba en línea. Podemos emular esta conducta al especificar un dominio de conmutación desordenado, irrestringido de un miembro.
- Dominio restringido: los servicios vinculados al dominio pueden ejecutarse únicamente en miembros de clúster que también son miembros de un dominio de conmutación. Si no hay miembros de conmutación disponibles, el servicio es puesto en el estado detenido. En un clúster con varios miembros, el uso de un dominio de conmutación restringido puede facilitar la configuración de un servicio de clúster (tal como un httpd), el cual requiere configuración idéntica en todos los miembros que ejecutan el servicio. En lugar de configurar el clúster completo para ejecutar el servicio de clúster, configure únicamente los miembros en el domino de conmutación restringido que usted asocia con el servicio de clúster.

- Dominio irrestringido: La conducta predeterminada, los servicios vinculados a este dominio pueden ejecutarse en todos los miembros de clúster, pero se ejecutan en un miembro de dominio siempre y cuando haya uno disponible. Esto significa que si un servicio está ejecutándose fuera del dominio y un miembro del dominio se conecta, el servicio migrará a ese miembro, a menos que se haya establecido la no recuperación.

- Dominio ordenado: El orden especificado en la configuración dicta el orden de preferencia de los miembros dentro del dominio. El miembro de más alto rango del dominio ejecutará el servicio cuando esté en línea. Esto significa que si el miembro A tiene un rango más alto respecto al miembro B, el servicio migrará a A si estaba ejecutándose en el miembro B, si el miembro A pasa de desconectado a conectado.

- Dominio desordenado: La conducta predeterminada, los miembros del dominio no tienen orden de preferencia; cualquier miembro puede ejecutar el servicio. Los servicios siempre migrarán a miembros del dominio de conmutación siempre que sea posible, no obstante, en un dominio desordenado.

- Recuperación: Los servicios en miembros de un dominio de conmutación ordenado deben volver al nodo que originalmente estaba ejecutándose antes de que el nodo fallara, lo cual es útil para evitar que los nodos que fallan con frecuencia eviten cambios de servicio frecuentes entre el nodo que falla y el nodo de conmutación.

El ordenamiento, la restricción y la no recuperación son indicadores que pueden combinarse en casi cualquier forma (por ejemplo, ordenado + restringido, desordenado + irrestringido, etc). Estas combinaciones afectan ambos servicios donde los servicios inician después de la formación del cuórum inicial y en los que los miembros de clúster se encargarán de los servicios en el evento que el servicio falle.

3.1.1. Ejemplos de conductas

Dado un clúster comprometido de este conjunto de miembros: {A, B, C, D, E, F, G}.

Dominio de conmutación ordenado, restringido {A, B, C}

Con no recuperación desconectada: Un servicio 'S' siempre se ejecutará en el miembro 'A' siempre que el miembro 'A' esté en línea y haya cuórum. Si todos los miembros de {A, B, C} están fuera de línea, el servicio no se ejecutará. Si el servicio se ejecuta en 'C' y 'A' pasa a en línea, el servicio migrará a 'A'.

Con no recuperación establecida: Un servicio 'S' se ejecutará en el miembro de clúster de prioridad más alta cuando se forme cuórum. Si todos los miembros de {A, B, C} están fuera de línea, el servicio no se ejecutará. Si el servicio se ejecuta en 'C' y 'A' pasa a en línea, el servicio permanecerá en 'C' a menos que 'C' falle, punto en el cual se recuperará a 'A'.

Dominio de conmutación desordenado, restringido {A, B, C}

Un servicio 'S' siempre se ejecutará si hay cuórum y si por lo menos un miembro de {A, B, C} está en línea. Si otro miembro del dominio pasa a en línea, el servicio no se reubicará.

Dominio de conmutación ordenado, irrestringido {A, B, C}

Con no recuperación sin configurar: Un servicio 'S' se ejecutará cuando haya cuórum. Si un miembro de dominio de conmutación está en línea, el servicio se ejecutará en el miembro de prioridad más alta, de lo contrario, se elegirá un miembro del clúster para ejecutar el servicio. Es decir, el servicio se ejecutará en 'A' siempre que 'A' esté en línea, seguido por 'B'.

Con no recuperación configurada: Un servicio 'S' se ejecutará cuando haya cuórum. Si un miembro de dominio de conmutación está en línea en formación de cuórum, el servicio se ejecutará en el

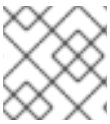
miembro de prioridad más alta del dominio de conmutación. Es decir que si 'B' está en línea (pero 'A' no lo está), el servicio se ejecutará en 'B'. Si en un punto más adelante, 'A' se une al clúster, el servicio no se reubicará en 'A".

Dominio de conmutación desordenado, irrestringido {A, B, C}

También denominado como un "Conjunto de miembros preferido". cuando uno o más miebros del dominio de conmutación están en línea el servicio se ejecutará en un miembro en línea no específico del dominio de conmutación. Si otro miembro de dominio de conmutación se tranfiere a en línea, no se reubica el servicio.

3.2. POLÍTICAS DE SERVICIOS

RGManager tiene tres políticas de servicios que el administrador puede personalizar según las necesidades del servicio.



NOTA

Estas políticas también aplican a los recursos de máquinas virtuales.

3.2.1. Política de inicio

RGManager inicia de forma predeterminada todos los servicios en el arranque de rgmanager y cuando hay un cuórum presente. Los administradores pueden modificar esta conducta.

- autostart (predeterminado) - inicia el servicio cuando rgmanager arranca y se forma un cuórum. Si se establece a '0', el clúster no iniciará el servicio y en su lugar lo dejará en estado inhabilitado.

3.2.2. Política de recuperación.

La política de recuperación es la acción predeterminada que rgmanager sigue cuando un servicio falla en un nodo determinado. Hay tres opciones disponibles, las cuales se definen en la lista a continuación.

- restart (predeterminado) - reinicia el servicio en el mismo nodo. Si no hay otra política de recuperación especificada, se utiliza la política de recuperación. Si el reinicio falla, rgmanager retrocede para reubicar el servicio.
- relocate - Trata de iniciar el servicio en otro nodo en el clúster. Si otros nodos no pueden reiniciar el servicio, el servicio será puesto en el estado detenido.
- disable - No hace nada. Establece el servicio a un estado inhabilitado.
- restart-disable - Intenta reiniciar el servicio, en su lugar. Si el reinicio falla, establece el servicio en estado inhabilitado.

3.2.3. Extensiones de políticas de reinicio

Cuando se utiliza la política de reinicio debe especificar también el máximo número de reinicios, que se pueden presentar en el mismo nodo en un tiempo determinado. Hay dos parámetros disponibles para servicios denominados max_restarts y restart_expire_time que controlan esto.

El parámetro max_restarts es un entero que especifica el número de reinicios antes de rendirse y reiniciar el servicio en otro host en el clúster.

El parámetro restart_expire_time le indica a rgmanager el tiempo para recordar un evento de reinicio.

El uso de los dos parámetros crea una ventana corrediza para el número de reinicios tolerados en una cantidad de tiempo determinado. Por ejemplo:

```
<service name="myservice" max_restarts="3" restart_expire_time="300" ...>
  ...
</service>
```

La tolerancia del servicio mencionado arriba es de 3 reinicios en 5 minutos. En el cuarto fallo de servicio en 300 segundos, rgmanager no reiniciará el servicio, en su lugar reubicará el servicio en otro host disponible en el clúster.

NOTA

Debe especificar ambos parámetros a la vez; el uso de cualquiera de los parámetros es indefinido.

3.3. ÁRBOLES DE RECURSOS - FUNDAMENTOS/DEFINICIONES

Lo siguiente ilustra la estructura de un árbol de recursos con la correspondiente lista que define cada área.

```
<service name="foo" ...>
    <fs name="myfs" ...>
        <script name="script_child"/>
    </fs>
    <ip address="10.1.1.2" .../>
</service>
```

- El árbol de recursos son representaciones XML de recursos, atributos, relaciones padre, hijo y hermanos. La raíz de un recurso es casi siempre un tipo especial de recursos llamado servicio. El árbol de recursos, el grupo de recursos, y el servicio suelen ser intercambiables en Wiki. Desde la perspectiva de rgmanager, un árbol de recursos es una unidad atómica. Todos los componentes de un árbol de recursos se inician en el mismo nodo de clúster.
- fs:myfs e ip:10.1.1.2 son hermanos
- fs:myfs es el padre del script:script_child
- script:script_child es el hijo de fs:myfs

3.3.1. Relaciones padre, hijo, dependencias y orden de inicio

Las reglas para relaciones padre e hijo en el árbol de recursos son bastante simples:

- Los padres se inician antes que los hijos
- Todos los hijos deben detenerse (limpiamente) para que el padre pueda parar.
- Se podría decir que el recurso de hijo depende del recurso del padre
- Para que un recurso pueda considerarse saludable, todos los hijos dependientes deben tener buena salud.

3.4. OPERACIONES DE SERVICIO Y ESTADOS

Las siguientes operaciones aplican a ambos servicios y a las máquinas virtuales, a excepción de la operación de migración, la cual únicamente funciona en máquinas virtuales.

3.4.1. Operaciones de servicio

Las operaciones de servicios son comandos disponibles que el usuario puede llamar para aplicar una de las cinco acciones definidas en la lista a continuación:

- enable — Inicia el servicio, opcionalmente en el destino preferido según las reglas de dominio de conmutación. En ausencia de un host local en donde se ejecuta clusvcadm, iniciará el servicio. Si el inicio original falla, el servicio se comportará como si se hubiese solicitado una operación reubicada (Ver abajo). Si la operación tiene éxito, el servicio se establecerá en el estado iniciado.

- disable — Detiene el servicio y lo pasa al estado inhabilitado. Esto solamente se permite cuando el servicio está en el estado fallido.

- relocate — Desplaza el servicio a otro nodo. También puede especificar un nodo preferido para recibir el servicio, pero la incapacidad del servicio para que se ejecute en ese host (por ejemplo, si no se puede iniciar el servicio o si el host está desconectado) no impide la reubicación, y se elige otro nodo. Rgmanager intenta iniciar el servicio en cada nodo del clúster admisible. Si ningún nodo de destino admisible en el clúster inicia el servicio, la reubicación falla y el servicio intenta reiniciarse en el propietario original. Si el propietario original no puede reiniciar el servicio, el servicio pasa al estado Detenido.

- stop — Detiene el servicio y lo pasa al estado detenido.

- migrate — Migra una máquina virtual a otro nodo. El administrador debe especificar un nodo de destino. Según la falla, si no puede migrar, la máquina virtual puede resultar en el estado fallido o en el estado iniciado en el propietario original.

3.4.1.1. La operación `freeze`

RGManager puede congelar servicios. Al hacerlo permite a los usuarios actualizar rgmanager, CMAN, o cualquier otro software en el sistema mientras minimiza el plazo de espera de los servicios de rgmanager.

Permite el mantenimiento de partes de servicios rgmanager. Por ejemplo, si tiene una base de datos y un servidor Web en un servicio rgmanager, puede congelar el servicio rgmanager, detener la base de datos, realizar mantenimiento, reiniciar la base de datos, y descongelar el servicio.

3.4.1.1.1. Conductas de servicios cuando están Congelados

- Las verificaciones de estatus se desactivan.

- Las operaciones de inicio se desactivan.

- Las operaciones de detenido se inhabilitan.

- La conmutación no ocurrirá (incluso si apaga al propietario del servicio)

IMPORTANTE

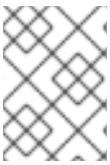
Si no sigue estos lineamientos puede hacer que los recursos se asignen a varios hosts.

- No debe detener todas las instancias de rgmanager cuando un servicio esté congelado a menos que planee reiniciar los hosts antes de reiniciar rgmanager.
- No debe descongelar un servicio hasta que el propietario reportado del servicio reconecte el clúster y reinicie el rgmanager.

3.4.2. Estados de servicios

La siguiente lista define los estados de servicios administrados por RGManager.

- disabled — El servicio permanecerá en el estado inhabilitado hasta que un administrador reactive el servicio o el servicio o el clúster pierdan cuórum (en el momento en que el parámetro autostart sea evaluado). El administrador puede habilitar el servicio desde este estado.
- failed — el servicio se presume muerto. Este estado se presenta cuando falla la operación para detener el recurso. El administrador debe verificar si hay recursos sin asignar (sistemas de archivos montados, pro ejemplo) antes de emitir una petición de inhabilitar, La única acción que puede tomar lugar desde este estado es inhabilitado.
- stopped — Cuando está en estado Detenido, el servicio será evaluado para iniciar después del siguiente servicio o transición de nodo. Esta es una medida muy temporal. Un administrador puede inhabilitar o habilitar el servicio desde este estado.
- recovering — El clúster trata de recuperar el servicio. El administrador puede desactivar el servicio para evitar la recuperación si se desea.
- started — Si la verificación del estatus de un servicio falla, recupérelo según la política de servicios. Si el host que está ejecutando el servicio falla, recupérelo con el dominio de conmutación y las reglas de servicio exclusivo. El administrador puede reubicar, detener, inhabilitar y (con máquinas virtuales) migrar el servicio desde este estado.



NOTA

Otros estados, tales como **`starting`** y **`stopping`** son estados de transición del estado **`started`**.

3.5. CONDUCTAS DE MÁQUINAS VIRTUALES

RGManager maneja máquinas virtuales un poco diferente a los otros servicios.

3.5.1. Operaciones normales

Las máquinas virtuales administradas por rgmanager deben ser administradas mediante clusvcadm u otra herramienta conocedora del clúster. La mayoría de las conductas son comunes con los servicios normales. Esto incluye:

- Iniciando (habilitando)
- Deteniendo (inhabilitando)

- Monitorización de estatus
- Reubicación
- Recuperación

Para obtener más información sobre los servicios virtuales disponibles, consulte el Capítulo 7, *Virtualización y alta disponibilidad*.

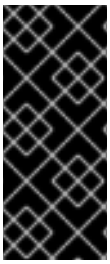
3.5.2. Migración

Además de las operaciones de servicios normales, las máquinas virtuales soportan una conducta que no recibe asistencia por otros servicios: migración. La migración minimiza el tiempo de inactividad de las máquinas virtuales al reducir el requerimiento de Inicio/Detención para cambiar el sitio de la máquina virtual dentro de un clúster.

Hay dos tipos de migración por rgmanager que son seleccionadas según la máquina virtual por el atributo de migración.

- live (predeterminado) — la máquina virtual continúa ejecutándose mientras la mayoría de su contenido de memoria se copia al host de destino. Esto minimiza el inaccesibilidad de la máquina virtual (por lo general de menos de 1 segundo) a expensas del rendimiento de una MV y la cantidad total de tiempo durante la migración expense of performance of the VM during the migration and total amount of time it takes for the migration to complete.
- pause - la máquina virtual se congela en memoria cuando el contenido de memoria se copia al host de destino. Esto minimiza la cantidad de tiempo que toma una máquina virtual en en completar la migración.

El estilo de migración que utilice depende de la disponibilidad y requerimientos de rendimiento. Por ejemplo, una migración puede significar segundos de rendimiento degradado y 1 segundo para no disponibilidad completa, mientras que una migración de pausa puede significar 8 segundos de no disponibilidad completa y ningún otro rendimiento degradado.



IMPORTANTE

Una máquina virtual puede ser un componente de servicio, pero al hacerlo, inhabilita todas las formas de migración y la mayoría de las funcionalidades de conveniencia de abajo.

Además, el uso de migración con KVM requiere configuración cuidadosa de ssh.

3.5.3. Características de máquinas virtuales RGManager

La sección a continuación lista las varias formas que RGManager facilita el uso de máquinas virtuales.

3.5.3.1. Seguimiento de máquina virtual

Iniciar una máquina virtual con **`clusvcadm`** si la MV ya está ejecutándose, hará que rgmanager busque el clúster para la MV y la marque como **`started`** siempre y cuando la encuentre.

Los administradores que accidentalmente migran a una máquina virtual (MV) con herramientas non-cluster, tales como **`virsh`** harán que rgmanager busque el clúster para MV y marque la MV como **`started`** siempre que se encuentre.

NOTA

Si la máquina virtual está ejecutándose en varios sitios, RGManager no le advierte.

3.5.3.2. Soporte de dominio transitorio

Rgmanager ofrece soporte a máquinas virtuales transitorias que tienen soporte de libvirt. Esto permite a rgmanager crear y retirar máquinas virtuales sobre la marcha, lo cual ayuda a reducir la posibilidad de inicios dobles de máquinas virtuales debido al uso de herramientas non-cluster.

El soporte a máquinas virtuales transitorias también le permite almacenar los archivos de descripción libvirt XML en un sistema de archivos en clúster para que no tenga que mantener manualmente **`/etc/libvirt/qemu`** en sincronía en todo el clúster.

3.5.3.2.1. Características de administración

La adición o remoción de una máquina virtual desde cluster.conf no iniciará o detendrá la MV; simplemente hará que rgmanager inicie o detenga prestando atención a la MV.

Recuperación (ir al nodo preferido) se realiza al usar migración para minimizar el tiempo de inactividad.

3.5.4. Conductas no manejables

Las siguientes condiciones y acciones de usuario no tienen soporte en RGManager.

- El uso de herramientas non-cluster-aware (por ejemplo, virsh o xm) para manipular un estado o configuración de máquina virtual cuando el clúster administra la máquina virtual. Verificar el estado de la máquina, es buena idea. (ej. virsh list, virsh dumpxml).
- Migrar una máquina virtual administrada de clúster a un nodo non-cluster o un nodo en el clúster que no ejecute rgmanager. Rgmanager iniciará la máquina virtual en la ubicación anterior, haciendo que dos instancias de máquina virtual se ejecuten, lo cual resulta en la corrupción del sistema de archivos.

3.6. ACCIONES DE RECURSOS

RGManager espera los siguiente valores de retorno de los agentes de recursos:

- start - iniciar el recurso
- stop - detener el recurso
- status - verificar el estatus del recurso
- metadata - reportar los metadatos OCF, RA y XML

3.6.1. Valores de retorno

OCF tiene una amplitud de rango de retorno de códigos de retorno para la operación de monitorización, pero puesto que rgmanager llama estatus, depende exclusivamente de los códigos de retorno en estilo SysV.

0 - éxito

Se detiene tras varias paradas cuando no está ejecutándose debe retornar éxito

Inicia tras varios inicios cuando se ejecuta debe retornar éxito

no cero - falla

Si la operación de detención nunca retorna un valor de no cero, el servicio entra en el estado fallido y debe ser recuperado manualmente.

CAPÍTULO 4. CERCADO

El cercado es la desconexión de un nodo desde el almacenamiento compartido del clúster. Este proceso corta la E/S desde y al almacenamiento compartido asegurando así la integridad de los datos. La infraestructura de clúster ejecuta el proceso de aislamiento a través del demonio de cercado **fenced**.

Cuando CMAN determina que el nodo ha fallado, comunica al otro clúster que los componentes de infraestructura de clúster han fallado. Cuando se notifica la falla al comando **fenced**, encierra el nodo que ha fallado. Otros componentes de infraestructura de clúster determinan las acciones a seguir — es decir, realizan cualquier recuperación que sea necesaria. Por ejemplo, cuando se notifica a DLM y GFS2 sobre la falla del nodo, suspenden la actividad hasta que detectan que el comando **fenced** ha completado el cercado del nodo fallido. Tras confirmación de que el nodo fallido ha sido cercado, DLM y GFS2 realizan la recuperación. DLM abre cerrojos del nodo fallido y GFS2 recupera el diario del nodo fallido.

El programa de aislamiento determina el método de cercado a utilizar desde el archivo de configuración de clúster. Hay dos elementos claves del archivo de configuración de clúster que definen el método de cercado: el agente y el dispositivo de cercado. El programa de cercado hace una llamada al agente de cercado especificado en el archivo de configuración del clúster. El agente de cercado, a su vez, aísla el nodo a través del dispositivo de cercado. Una vez el proceso de aislamiento ha sido completado, el programa de aislamiento notifica al administrador de clúster.

La adición de alta disponibilidad proporciona una variedad de métodos de cercado:

- Aislamiento de energía — Un método de cercado que utiliza un controlador de energía para apagar el nodo que no funciona.
- Cercado de almacenamiento — Un método de cercado que inhabilita el puerto de canal de fibra que conecta el almacenamiento a un nodo que no funciona.
- Otros métodos de cercado — Hay otros métodos de cercado que desactivan la E/S o apagan el nodo que no funciona; incluidos IBM Bladecenters, PAP, DRAC/MC, HP ILO, IPMI, IBM RSA II y otros.

La Figura 4.1, “Ejemplo de cercado de energía” muestra un ejemplo de cercado de energía. En el ejemplo, el programa de cercado en el nodo A hace que el controlador de energía apague el nodo D. La Figura 4.2, “Ejemplo de cercado de almacenamiento” muestra un ejemplo de cercado de almacenamiento. En el ejemplo, el programa de cercado en nodo A hace que el interruptor de Canal de fibra inhabilite el puerto para el nodo D, al desconectar el nodo D del almacenamiento. .

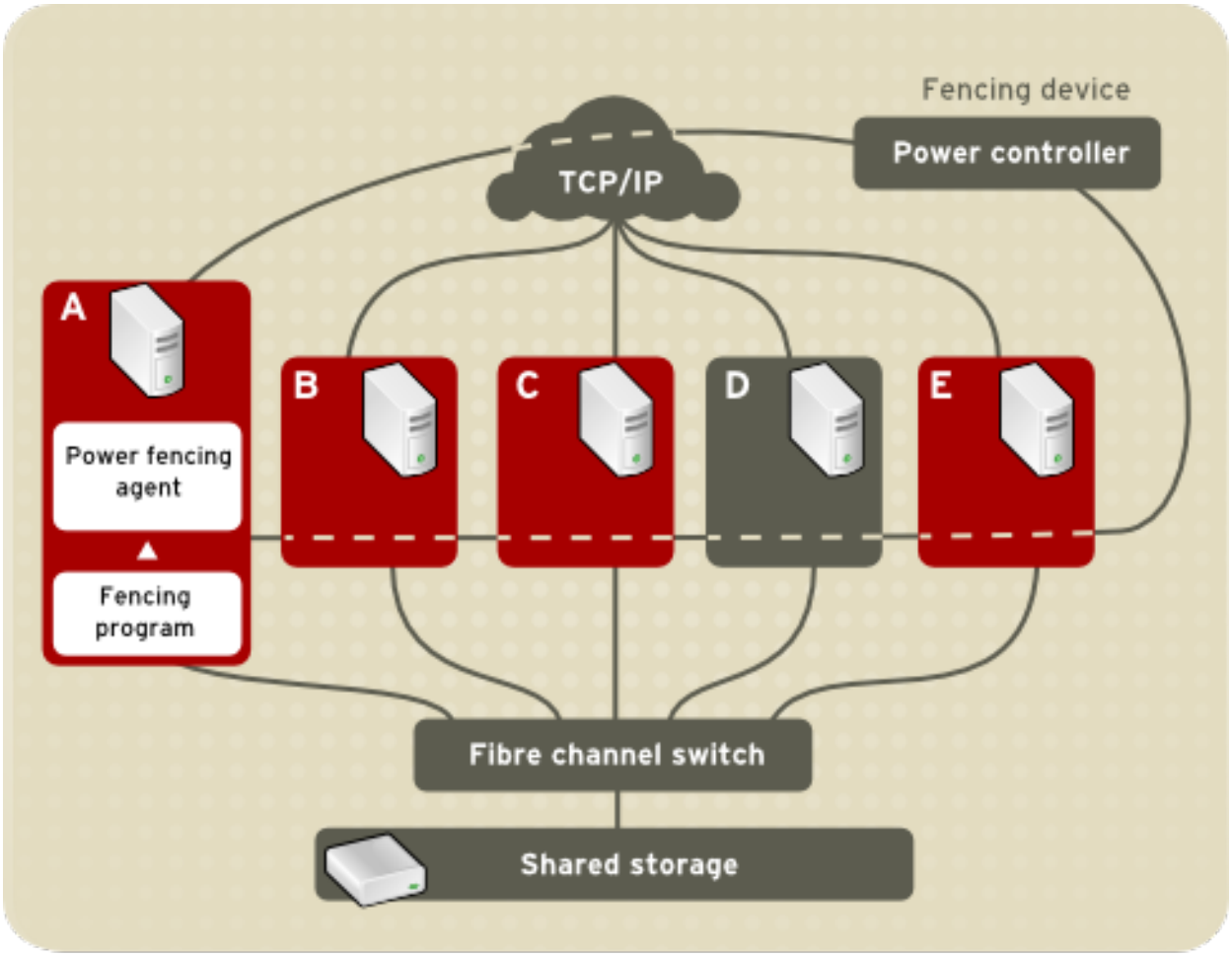


Figura 4.1. Ejemplo de cercado de energía

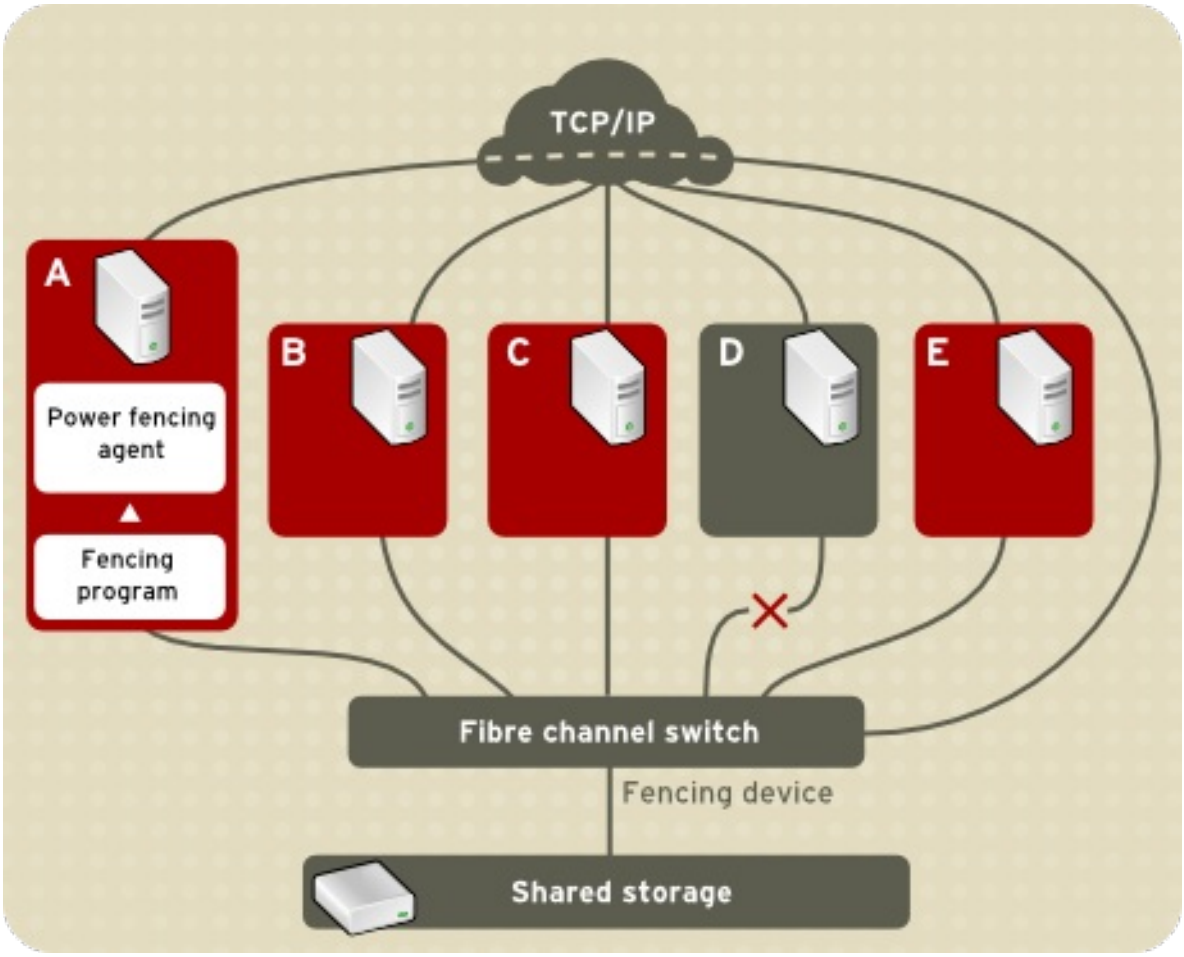


Figura 4.2. Ejemplo de cercado de almacenamiento

Para especificar un método de cercado se debe editar el archivo de configuración para asignar el nombre del método de cercado, el agente de cercado y el dispositivo de cercado para cada nodo en el clúster.

La forma en que se especifica un método de cercado depende de si el nodo tiene abastecimiento de energía doble o rutas múltiples de almacenamiento. Si un nodo tiene abastecedores de energía doble, entonces el método de cercado para el nodo debe especificar al menos dos dispositivos de cercado — un dispositivo de cercado para cada abastecedor de energía (ver la Figura 4.3, “Cercado de un nodo con abastecedores de energía doble”). Igualmente, si un nodo tiene múltiples rutas a almacenamiento de Canal de fibra, entonces el método de cercado para el nodo debe especificar un dispositivo de cercado para cada ruta al almacenamiento de Canal de fibra. Por ejemplo, si un nodo tiene dos rutas al almacenamiento de Canal de fibra, el método de cercado debe especificar dos dispositivos de cercado — uno para cada ruta al almacenamiento de Canal de fibra (ver la Figura 4.4, “Cercado de un nodo con conexiones de canal de fibra doble”).

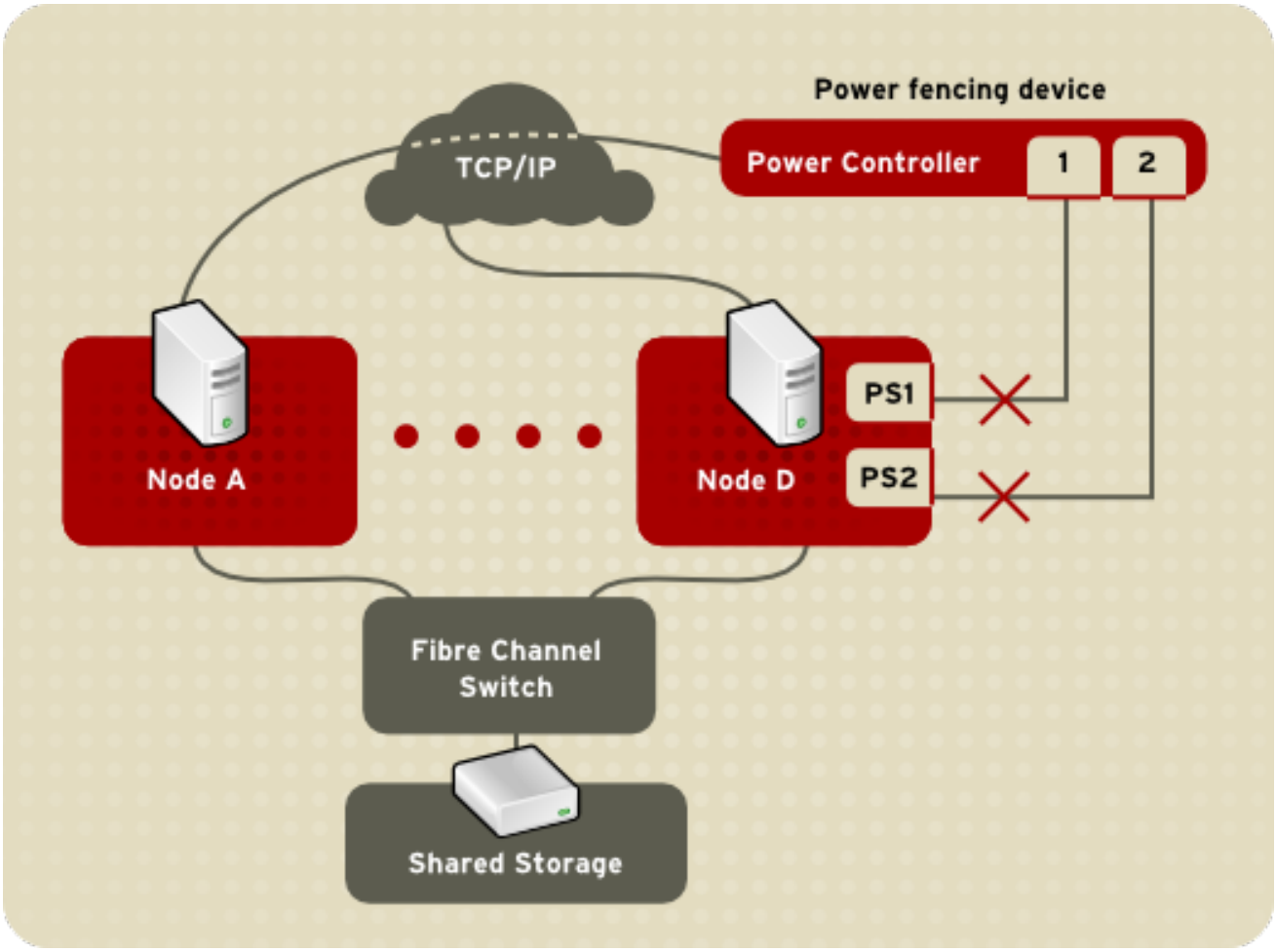


Figura 4.3. Cercado de un nodo con abastecedores de energía doble

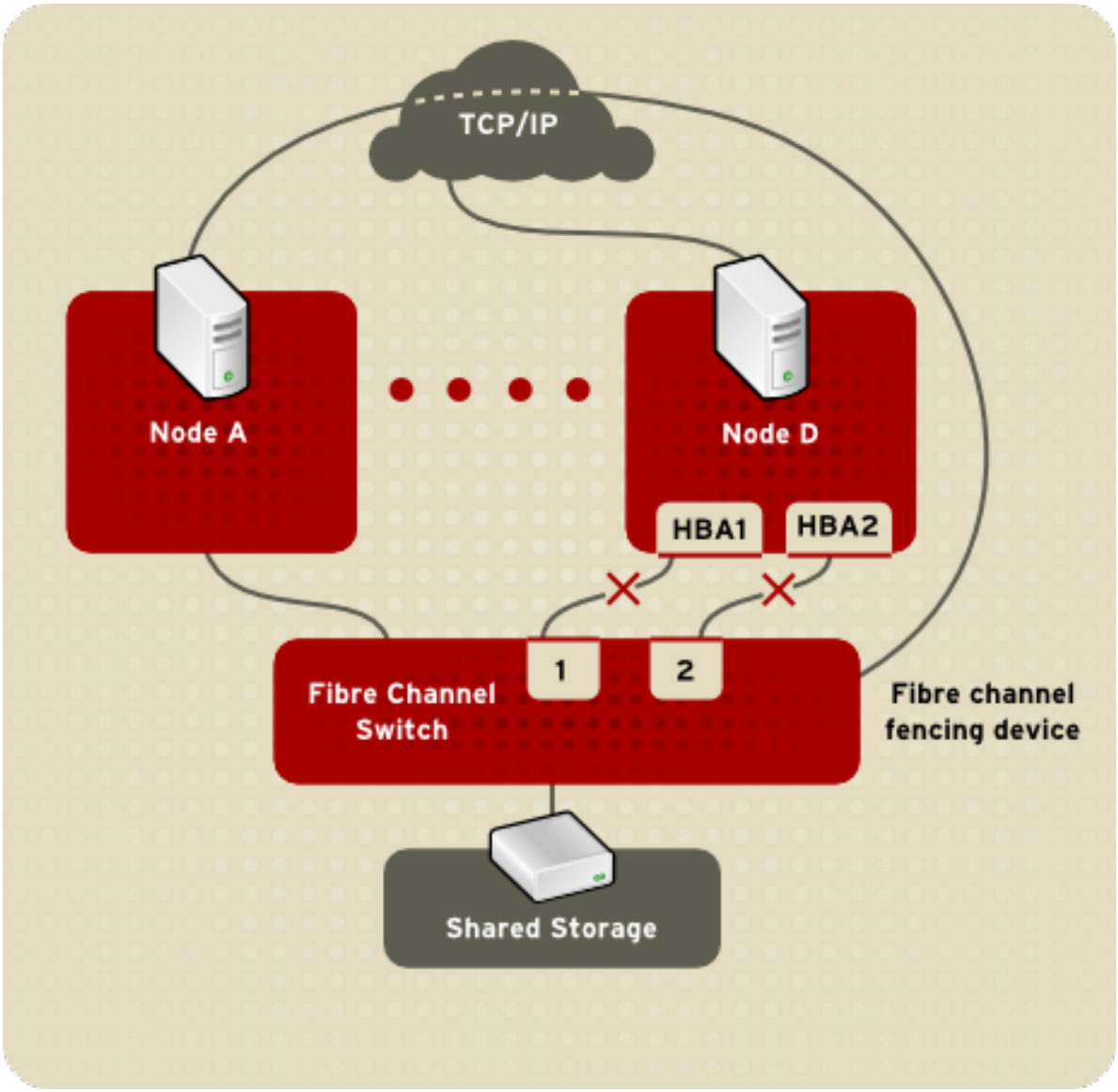


Figura 4.4. Cercado de un nodo con conexiones de canal de fibra doble

Puede configurar un nodo con uno o más métodos de cercado. Cuando se utiliza más de un método de cercado, se utilizan en *cascada*, en el orden de prioridad dado en el archivo de configuración de clúster. Si un nodo falla, es aislado mediante el primer método de cercado especificado en el archivo de configuración de clúster para ese nodo. Si el primer método de cercado no funciona, se utiliza el siguiente método de cercado especificado para ese nodo. Si ninguno de los métodos de cercado funciona, el primer método de cercado se ejecutará nuevamente y este bucle continúa hasta que el nodo haya sido cercado.

Para obtener información más detallada sobre configuración de dispositivos de cercado, consulte el capítulo correspondiente en el manual de *Administración de clúster.*

CAPÍTULO 5. ADMINISTRACIÓN DE CERROJO

La administración de cerrojo es un servicio de infraestructura de clúster común que proporciona un mecanismo para otros componentes de infraestructura de clúster para sincronizar su acceso a recursos compartidos. En un clúster de Red Hat, DLM (Gestor de cerrojo distribuido) es el gestor de cerrojo.

Un gestor de cerrojo es un policía de trafico que controla el acceso a los recursos en un clúster, tales como acceso a un sistema de archivos GFS. Es necesario porque sin un gestor de cerrojo, no habrá control sobre el acceso a su almacenamiento compartido, y los nodos en el clúster dañarían los datos de cada uno.

DLM es un gestor de cerrojo distribuido que se ejecuta en cada nodo de clúster; la administración de cerrojo se distribuye a través de todos los nodos en el clúster. GFS2 y CLVM emplean cerrojos del gestor de cerrojo. GFS2 utiliza cerrojos del gestor de cerrojo para sincronizar el acceso a metadatos del sistema de archivos (en almacenamiento compartido). CLVM emplea cerrojos del gestor de cerrojo para sincronizar actualizaciones para volúmenes de LVM y grupos de volúmenes (también en almacenamiento compartido). Además, `rgmanager` usa DLM para sincronizar los estados de servicios.

5.1. MODELO DE CERRAMIENTO DE DLM

El modelo de cerramiento DLM proporciona una serie de modos de cerrojo de ejecución tanto asíncrona como sincrónica. Una aplicación adquiere un cerrojo en un recurso de cerrojos. Existe una relación de uno o muchos entre recursos y cerrojos: un recurso de cerrojo individual puede tener múltiples cerrojos asociados a este.

Un recurso de cerrojos puede corresponder al objeto real, tal como un archivo, una estructura de datos, una base de datos o una rutina ejecutable, pero no necesariamente debe corresponder a alguna de ellas. El objeto que usted asocie con el recurso de cerrojos determina la granularidad del cerrojo. Por ejemplo, encerrar toda una base de datos se considera un cerramiento en granularidad gruesa. Encerrar cada elemento en una base de datos se considera un cerramiento en una granularidad fina.

El modelo de cerramiento DLM soporta:

- Seis modos de cerramiento que restringen en aumento el acceso a un recurso
- Promoción y degradación de cerrojos mediante conversión
- Terminación sincrónica
- Terminación asíncrona
- Datos globales mediante bloques de valor de cerrojo

El DLM provee sus propios mecanismos para soportar sus funcionalidades de cerramiento, tales como la comunicación internodal para manejar el tráfico de cerrojos y los protocolos de recuperación para remasterizar cerrojos tras una falla de nodos o para migrar cerrojos cuando un nodo se une al clúster. No obstante, el DLM no proporciona mecanismos para administrar en realidad el clúster. Por lo tanto, el DLM espera operar en un clúster junto con otro entorno de infraestructura de clúster que proporcione los siguientes requisitos mínimos:

- El nodo es una parte de un clúster.
- Todos los nodos concuerdan en membresía de clúster y cuórum.
- Una dirección IP debe comunicarse con el DLM en un nodo. Normalmente el DLM usa

comunicaciones internodales de TCP/IP que la restringen a una sola dirección IP por nodo (aunque puede hacerse más redundante mediante el dispositivo de vinculación). El DLM puede ser configurado para usar SCTP como su transporte internodal que permite múltiples direcciones IP por nodo.

El DLM funciona con cualquier entorno de infraestructura de clúster que proporciona los requerimientos mínimos listados arriba. La selección de un entorno de código abierto o cerrado depende del usuario. Sin embargo, la mayor limitación de DLM es la cantidad de pruebas realizadas con entornos diferentes.

5.2. ESTADOS DE CERROJO

Un estado de cerrojo indica el estatus actual de una solicitud de cerrojo. Un cerrojo siempre está en alguno de los tres estados:

- Aprobada — La solicitud de cerrojo fue aprobada y alcanzó el modo requerido.
- Convirtiendo — Un cliente intentó cambiar el modo de cerrojo y el nuevo modo no es compatible con el cerrojo existente.
- Bloqueado — La solicitud de un nuevo cerrojo podría no ser aprobada debido a que existen cerrojos en conflicto.

El estado de cerrojo es determinado por el modo solicitado y los modos de los otros cerrojos en el mismo recurso.

CAPÍTULO 6. HERRAMIENTAS DE ADMINISTRACIÓN Y CONFIGURACIÓN

El archivo de configuración de clúster, **`/etc/cluster/cluster.conf`** especifica la configuración de la adición de alta disponibilidad. El archivo de configuración es un archivo XML que describe las siguientes características de clúster:

- Nombre de clúster — Especifica el nombre de clúster, el nivel de revisión de archivo de configuración de clúster y las propiedades de tiempo de cercado básicas utilizadas cuando un nodo se vincula a un clúster o es cercado desde el clúster.
- Clúster — Especifica cada nodo del clúster: nombre de nodo, ID de nodo, número de votos de cuórum y método de cercado para dicho nodo.
- Dispositivo de vallas — Especifica los dispositivos de vallas en el clúster. Los parámetros varían según el tipo de dispositivo de vallas. Por ejemplo, para un controlador de energía utilizado como un dispositivo de vallas, la configuración de clúster define el nombre del controlador de energía, la dirección IP, el nombre de usuario y la contraseña.
- Recursos administrados — Especifica los recursos requeridos para crear servicios de clúster. Los recursos administrados incluyen la definición de dominios de conmutación, los recursos (por ejemplo una dirección IP) y los servicios. Los recursos administrados definen los servicios de clúster y la conducta de conmutación de los servicios de clúster.

La configuración de clúster se valida automáticamente según el esquema del clúster en **`/usr/share/cluster/cluster.rng`** durante el tiempo de inicio y cuando la configuración se vuelve a cargar. También, puede validar una configuración de clúster en cualquier momento con el comando **`ccs_config_validate`**.

Un esquema anotado está disponible a la vista en **`/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html`** (Por ejemplo: **`/usr/share/doc/cman-3.0.12/cluster_conf.html`**).

Validación de configuración chequea los siguientes errores básicos:

- Validez XML — Verifica si archivo de configuración es un archivo XML válido.
- Opciones de configuración — Verifica si las opciones (elementos XML y atributos) son válidas.
- Valores de opción — Verifica si las opciones contienen datos válidos (limitados).

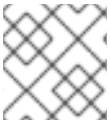
6.1. HERRAMIENTAS DE ADMINISTRACIÓN DE CLÚSTER

El manejo de Red Hat High Availability Add-On consiste en usar herramientas de configuración para especificar la relación entre los componentes de clúster. Las siguientes herramientas de configuración de clúster están disponibles con dicho complemento:

- **Conga** — Es una interfaz de usuario global para instalar, configurar y administrar Red Hat High Availability Add-On. Consulte *Configuring and Managing the High Availability Add-On* para obtener información acerca de cómo configurar y administrar la adición de alta disponibilidad con **Conga**.
  - Luci — Es el servidor de aplicaciones que proporciona la interfaz de usuario para Conga. Permite a los usuarios manejar servicios de clúster y proporciona acceso a ayuda y documentación en línea cuando se necesite.

- Ricci — Es un demonio de servicios que maneja distribución de la configuración de clúster. Los usuarios pasan información de configuración mediante la interfaz Luci y la configuración se carga en corosync para distribución a nodos de clúster.

- A partir del lanzamiento de Red Hat Enterprise Linux 6.1, Red Hat High Availability Add-On ofrece soporte para el comando de configuración de clúster **ccs**, el cual permite al administrador crear, modificar, y ver el archivo de configuración de clúster cluster.conf. Consulte el manual de *Administración de clúster*, para obtener información sobre cómo configurar y administrar la Adición de alta disponibilidad con el comando **ccs**.



NOTA

`system-config-cluster` no está disponible en RHEL 6.

CAPÍTULO 7. VIRTUALIZACIÓN Y ALTA DISPONIBILIDAD

Varias plataformas de virtualización son compatibles con Red Hat Enterprise Linux 6 con los complementos de almacenamiento resistente y de alta disponibilidad. Hay dos usos compatibles para virtualización junto con Red Hat Enterprise Linux High Availability Add-on.

Se refiere a RHEL Cluster/HA que se ejecuta en hosts vacíos utilizables como plataformas de virtualización. En este modo puede configurar el gestor de recursos de clúster (rgmanager) para administrar las máquinas virtuales (huéspedes) como recursos de alta disponibilidad.

- Máquinas virtuales como recursos o servicios disponibles
- Clústeres de huéspedes

7.1. MÁQUINAS VIRTUALES COMO RECURSOS O SERVICIOS DISPONIBLES

RHEL HA y RHEV ofrecen máquinas virtuales de alta disponibilidad (HA). Debido a la coincidencia en funcionalidad, se debe tener cuidado al escoger el producto para que se ajuste a su caso de uso específico. Los siguientes lineamientos deben tenerse en cuenta al elegir entre RHEL HA y RHEV para proporcionar alta disponibilidad de Máquinas virtuales.

Para máquina virtual y conteo de hosts físicos

- Si un gran número de máquinas virtuales están hechas para alta disponibilidad de una cantidad grande de hosts físicos, el uso de RHEL puede ser la mejor solución, ya que tiene más algoritmos sofisticados para administrar la colocación de máquina virtual que tienen en cuenta cosas como CPU, memoria e información de carga.
- Si un pequeño número de máquinas virtuales está hecho para alta disponibilidad de un reducido número de hosts físicos, el uso de alta disponibilidad (HA) de RHEL, puede ser la mejor solución, ya que se requiere menos infraestructura adicional. La solución de RHEV más pequeña requiere 4 nodos: 2 para proporcionar alta disponibilidad para el servidor RHEVM y 2 para actuar como hosts de máquina virtual.
- No hay lineamientos estrictos para el número de hosts o máquinas virtuales que se considerarían como 'una gran cantidad'. Sin embargo, tenga en cuenta que el número máximo de hosts en un clúster de alta disponibilidad de RHEL es 16 y que ningún clúster con 8 o más hosts necesitará una revisión de arquitectura de Red Hat para determinar la compatibilidad.

Uso de máquina virtual:

- Si sus máquinas virtuales de alta disponibilidad (HA) que ofrecen servicios, proporcionan una infraestructura compartida, se puede utilizar HA RHEL o RHEV..
- Si necesita proporcionar alta disponibilidad (HA) para una pequeña serie de servicios importantes que se ejecutan dentro de máquinas virtuales, alta disponibilidad de RHEL o RHEV.
- Debería utilizar RHEV, si busca proporcionar infraestructura para aplicar un rápido aprovisionamiento de máquinas virtuales.
    - La alta disponibilidad de máquina virtual RHEV debe ser dinámica. La adición de nuevas máquinas virtuales a un 'clúster' de RHEV es factible y tiene total soporte.
    - La alta disponibilidad de máquina virtual RHEV no es para entornos altamente dinámicos.

Un clúster con una serie de máquinas virtuales fijas, debe configurarse y luego para el tiempo de vida del clúster no se recomienda agregar o retirar máquinas virtuales adicionales.

- Alta disponibilidad de RHEL no debe ser utilizada para proporcionar infraestructura para la creación de entornos-similares a cloud debido a la naturaleza estática de configuración de clúster como también el conteo relativamente bajo de máximo de nodos físicos(16 nodos)

RHEL 5 soporta dos plataformas de virtualización. Xen ha tenido soporte desde el lanzamiento de RHEL 5.0. Se introdujo en RHEL 5.4 KVM.

RHEL 6 solamente soporta KVM como plataforma de virtualización.

RHEL 5 AP Cluster ofrece soporte para KVM y Xen para uso en máquinas virtuales que son administradas por la infraestructura de clúster de host.

Alta disponibilidad de RHEL 6 soporta KVM para usar en máquinas virtuales que son administradas por la infraestructura de clúster de host.

La lista a continuación muestra los escenarios de implementación que tienen soporte de Red Hat actualmente:

- RHEL 5.0+ ofrece soporte para Xen junto con RHEL AP Cluster
- RHEL 5.4 introdujo soporte para máquinas virtuales KVM como recursos administrados en RHEL AP Cluster como una muestra previa de tecnología.
- RHEL 5.5+ aumenta el soporte para que las máquinas virtuales tengan soporte completo.
- RHEL 6.0+ soporta máquinas virtuales KVM como recursos de alta disponibilidad en RHEL 6 High Availability Add-On.
- RHEL 6.0+ no ofrece soporte para máquinas virtuales Xen con RHEL 6 High Availability Add-On, a partir de RHEL 6 Xen ya no recibe soporte.

NOTA

Para obtener más información y notas especiales sobre escenarios de implementación con soporte, consulte la siguiente entrada en Red Hat Knowledgebase:

https://access.redhat.com/kb/docs/DOC-46375

Los tipos de máquinas virtuales que se ejecutan como recursos administrados no importan. Cualquier huésped que tenga soporte ya sea para Xen o KVM en RHEL puede ser utilizada como huésped altamente disponible. Esto incluye las variantes de RHEL (RHEL3, RHEL4, RHEL5) y varias variantes de Microsoft Windows. Verifique la documentación RHEL para buscar las listas más recientes de sistemas operativos de huésped que reciben soporte en cada hipervisor.

7.1.1. Recomendaciones generales

- En RHEL 5.3 y anteriores, rgmanager utilizaba las interfaces Xen nativas para administrar los Xen domU (huéspedes). En RHEL 5.4 esto se cambió para usar libvirt para los hipervisores Xen y KVM a fin de proveer una interfaz consistente en ambos tipos de hipervisor. Aparte de este cambio de arquitectura hay numerosas correcciones de errores en RHEL 5.4 y 5.4.z, por lo tanto, antes de configurar los servicios administrados de Xen, se recomienda actualizar sus clústeres de host por lo menos a los paquetes RHEL 5.5 más recientes.

- Para servicios KVM administrados, actualícese a RHEL 5.5, ya que esta es la primera versión de RHEL en la que esta funcionalidad recibe soporte completo.

- Siempre verifique las erratas más recientes antes de implementar un clúster para asegurarse de que tiene las últimas correcciones para los problemas o errores conocidos.

- La mezcla de hosts de diferentes tipos de hipervisores no tiene soporte. El clúster de host debe ser todo Xen o basado en KVM.

- El hardware de hosts debe aprovisionarse de tal forma que pueda absorber los huéspedes reubicados desde otros hosts fallidos sin hacer que un host se exceda en memoria o CPU virtuales. Si hay fallas suficientes para producir exceso de memoria de las CPU virtuales, se puede producir una severa degradación del rendimiento y en potencia, una falla de clúster.

- El uso directo de herramientas XM o libvirt (virsh, virt-manager) para administrar máquinas virtuales (live migrate, stop. start) que están bajo el control de rgmanager no tienen soporte y no se recomienda debido a que evitaría la pila de administración de clúster.

- Cada MV debe ser de un ancho de clúster único, incluidas las MV locales únicamente o sin clúster. Libvirtd solo aplica los nombres únicos según el host. Si clona una MV a mano, debe cambiar el nombre del archivo de configuración de clon.

7.2. CLÚSTERES DE HUÉSPEDES

Se refiere a RHEL Cluster/HA que se ejecuta dentro de los huéspedes virtualizados en una variedad de plataformas de virtualización. En este caso RHEL Clustering/HA se utiliza principalmente para hacer que las aplicaciones que se ejecuten dentro de los huéspedes estén altamente disponibles. Este escenario es similar a la forma como RHEL Clustering/HA siempre ha utilizado hosts vacíos tradicionales. La diferencia es que el agrupamiento se ejecuta dentro de los huéspedes en su lugar.

La siguiente es una lista de plataformas de virtualización y el nivel de soporte actualmente disponible para ejecutar clústeres de huéspedes mediante alta disponibilidad (HA) de clústeres de RHEL. En la lista, RHEL 6 Guests abarca tanto la alta disponibilidad (agrupamiento de núcleos) como los complementos de almacenamiento resistente (GFS2, clvmd y cmirror).

- Hosts de RHEL 5.3+ Xen soportan totalmente los clústeres de huéspedes en ejecución donde los sistemas operativos de huéspedes también son RHEL 5.3 o superiores:

    - Clústeres de huéspedes Xen pueden usar fence_xvm o fence_scsi para cercado de huéspedes.

    - El uso de fence_xvm/fence_xvmd requiere que se esté ejecutando un clúster de hosts para ofrecer soporte. fence_xvmd y fence_xvm deben utilizarse como agentes de cercado en todos los huéspedes en clúster.

    - El almacenamiento compartido puede ser provisto, ya sea por dispositivos de bloque compartidos respaldados por iSCSI o Xen o por el almacenamiento de archivos de respaldo (imágenes crudas).

- Hosts RHEL 5.5+ KVM no soportan clústeres de huéspedes en ejecución.

- Hosts RHEL 6.1+ KVM no soportan clústeres de huéspedes en ejecución donde los sistemas operativos de huésped son huéspedes RHEL 6.1+ o RHEL 5.6+. Los huéspedes de RHEL 4 no tienen soporte.

    - La mezcla de nodos de clúster en vacío que están virtualizados es permitida.

  - Clústeres de huéspedes RHEL 5.6+ pueden usar fence_xvm o fence_scsi para cercado de huéspedes.
  - Clústeres de huéspedes RHEL 6.1+ pueden usar ya sea fence_xvm (en el paquete **fence-virt** package) o fence_scsi para cercado de huéspedes.
  - Los RHEL 6.1+ KVM Hosts deben usar fence_virtd si el clúster de huéspedes utiliza fence_virt o fence_xvm como el agente de vallas. Si el clúster de huéspedes utiliza fence_scsi, no requerirá fence_virtd en los hosts.
  - fence_virtd puede operar en tres modos:
    - No se permite el modo autónomo en el que el mapeo de host a huésped tiene codificación dura y migración en vivo de los huéspedes.
    - Uso del servicio Openais Checkpoint para realizar el seguimiento a migraciones en vivo de huéspedes en clúster. Para ello, se requiere que un clúster de host esté ejecutándose.
    - Uso de Infraestructura de administración Qpid (QMF) provista por el paquete libvirt-qpid. Utiliza QMF para hacer seguimiento a migraciones de huéspedes sin requerir que todo un clúster de host esté presente.
  - El almacenamiento compartido puede ser provisto, ya sea por dispositivos de bloque compartidos respaldados por iSCSI o KVM o por el almacenamiento de archivos de respaldo (imágenes crudas).

- Las versiones 2.2+ y 3.0 de Red Hat Enterprise Virtualization Management (RHEV-M) soportan actualmente huéspedes en clúster RHEL 5.6+ y RHEL 6.1+ .
  - Los clústeres de huéspedes deben ser homogéneos (ya sean todos los huéspedes RHEL 5.6+ o todos los huéspedes RHEL 6.1+).
  - La mezcla de nodos de clúster en vacío que están virtualizados es permitida.
  - El cercado es provisto por fence_scsi en RHEV-M 2.2+ y por fence_scsi y fence_rhevm en RHEV-M 3.0. El cercado recibe soporte mediante fence_scsi como se describe a continuación:
    - El uso de _scsi con almacenamiento iSCSI es limitado para servidores iSCSI que soportan reservaciones persistentes SCSI 3 con los comandos preempt-y -abort. No todos los servidores iSCSI soportan esta funcionalidad. Verifique con el proveedor de almacenamiento si su servidor cumple con el soporte de reservaciones persistentes SCSI 3. Observe que el servidor iSCSI distribuido con Red Hat Enterprise Linux no soporta actualmente reservaciones persistentes SCSI 3 , por lo tanto, no es apto para usar con fence_scsi.

- VMware vSphere 4.1, VMware vCenter 4.1, VMware ESX y ESXi 4.1 soportan clústeres de huéspedes en ejecución cuando los sistemas operativos de huésped son RHEL 5.7+ o RHEL 6.2+. La versión 5.0 de VMware vSphere, vCenter, ESX y ESXi también tienen soporte; no obstante, debido al esquema incompleto de WDSL provisto en el lanzamiento inicial de Vmware vSphere 5.0, la herramienta fence_vmware_soap no funciona en la instalación predeterminada. Consulte nuestra base de conocimientos en https://access.redhat.com/knowledge/ para obtener el procedimiento actualizado para corregir este problema.
  - Los clústeres de huéspedes deben ser homogéneos (sean todos los huéspedes RHEL 5.7+ o todos los huéspedes RHEL 6.1+).

    - La mezcla de nodos de clúster en vacío que están virtualizados es permitida.
    - El agente fence_vmware_soap requiere VMware perl API de terceras partes. Este paquete de software debe descargarse desde el sitio de Web VMware e instalarse en los huéspedes en clúster de RHEL.
    - De modo alternativo, fence_scsi puede utilizarse para proveer cercado como se describe a abajo.
    - El almacenamiento compartido puede ser provisto por dispositivos de bloque compartidos crudos iSCSI o VMware.
    - El uso de clústeres de huéspedes VMware ESX recibe soporte, sea con fence_vmware_so_ap o fence_scsi.
- El uso de clústeres de huéspedes Hyper-V no recibe soporte, en este momento.

7.2.1. El uso de fence-scsi y almacenamiento compartido iSCSI

- En todos los entornos de virtualización de arriba, el almacenamiento de fence_scsi e iSCSI se pueden utilizar, en lugar del almacenamiento nativo compartido y los dispositivos nativos de vallas.
- fence_scsi se utiliza para proporcionar cercado de E/S para almacenamiento compartido provisto en iSCSI si el destino iSCSI soporta reservaciones persistentes SCSI 3 y los comandos preempt y abort. Verifique con el vendedor de almacenamiento para determinar si su solución iSCSI soporta la funcionalidad de arriba.
- El software de servidor iSCSI que se distribuye con RHEL no soporta reservaciones persistentes SCSI 3, por lo tanto, no puede utilizarse con fence_scsi. No obstante, es apto para ser utilizado como una solución de almacenamiento compartida junto con otros dispositivos de vallas tales como, fence_vmware o fence_rhevm.
- Para usar fence_scsi en todos los huéspedes, no se requiere un clúster de host (en RHEL 5 Xen/KVM y RHEL 6 KVM Host use casos)
- Si usa fence_scsi como agente de vallas, todo el almacenamiento compartido debe ser sobre iSCSI. No se permite la mezcla de iSCSI y almacenamiento compartido.

7.2.2. Recomendaciones generales

- Como se indicó anteriormente, antes de usar funcionalidades de virtualización se recomienda actualizar el host y los huéspedes a los paquetes más recientes de RHEL, puesto que se han realizado muchas correcciones y mejoras.
- La mezcla de plataformas (hipervisores) subyacentes a los clústeres de huéspedes, no recibe soporte. Todos los hosts subyacentes deben usar la misma tecnología de virtualización.
- No se ofrece soporte si se ejecutan todos los huéspedes en un clúster de huésped en un host físico único, ya que no se proporciona alta disponibilidad en el evento de una falla de un host único. Sin embargo, esta configuración puede utilizarse para propósitos de desarrollo o prototipo.
- Las mejores prácticas se incluyen a continuación:
    - No es necesario tener un host individual para cada huésped, pero esta configuración sí

proporciona la más alta disponibilidad, puesto que una falla de host solamente afecta un nodo individual en el clúster. Si tiene asignados 2 por 1 (dos huéspedes en un clúster único por un host físico), significa que la falla de host individual, resulta en fallo de dos huéspedes. Por lo tanto, se recomienda que en lo posible, se esté cerca de la asignación 1 por 1.

- La mezcla de clústeres de huéspedes independientes en el mismo conjunto de hosts físicos, actualmente no recibe soporte cuando se usan los agentes de vallas fence_xvm/fence_xvmd o fence_virt/fence_virtd.

- La mezcla de clústeres de huéspedes independientes en el mismo set de hosts físicos funciona si se utiliza almacenamiento _scsi + iSCSI o fence_vmware + VMware (ESX/ESXi y vCenter).

- La ejecución de huéspedes sin clúster en el mismo set de hosts físicos como un clúster de huéspedes recibe soporte, pero, puesto que los hosts se cercarán físicamente entre sí cuando se configura un clúster de hosts, los demás huéspedes también se terminarán durante la operación de cercado de host.

- El hardware de hosts debe aprovisionarse para evitar el sobrenvío de memoria o de CPU virtual. El sobrenvío de memoria o de CPU virtual, resultará en degradación de rendimiento. Si la degradación de rendimiento se torna crítica, el latido de clúster, podría afectarse, lo cual puede producir un fallo de clúster.

APÉNDICE A. HISTORIA DE REVISIONES

Revisión 1-15.3 **Mon Mar 2 2015** **Gladys Guerrero-Lozano**
Translated

Revisión 1-15.2 **Mon Mar 2 2015** **Gladys Guerrero-Lozano**
Translated

Revisión 1-15.1 **Mon Mar 2 2015** **Gladys Guerrero-Lozano**
Los archivos de traducción sincronizados con fuentes XML 1-15

Revisión 1-15 **Tue Dec 16 2014** **Steven Levine**
Actualización para implementar sort_order en la página de inicio de RHEL 6.

Revisión 1-13 **Wed Oct 8 2014** **Steven Levine**
Lanzamiento de disponibilidad general para Red Hat Enterprise Linux 6.6

Revisión 1-12 **Thu Aug 7 2014** **Steven Levine**
Lanzamiento Beta Red Hat Enterprise Linux 6.6

Revisión 1-11 **Fri Aug 1 2014** **Steven Levine**
Resuelve: #852720
Problemas menores de edición

Revisión 1-10 **Fri Jun 6 2014** **Steven Levine**
Borrador para Red Hat Enterprise Linux 6.6

Revisión 1-7 **Wed Nov 20 2013** **John Ha**
Lanzamiento para disponibilidad general de Red Hat Enterprise Linux 6.5

Revisión 1-4 **Mon Feb 18 2013** **John Ha**
Lanzamiento para disponibilidad general de Red Hat Enterprise Linux 6.4

Revisión 1-3 **Mon Jun 18 2012** **John Ha**
Actualización para alta disponibilidad de Red Hat Enterprise Linux 6.3

Revisión 1-2 **Fri Aug 26 2011** **John Ha**
Actualización para lanzamiento 6.2

Revisión 1-1 **Wed Nov 10 2010** **Paul Kennedy**
Lanzamiento inicial